universität wien

MASTERARBEIT/MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

A detailed look at the Adaptive Gradient Descent method

verfasst von / submitted by

Marcel Harrer, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2023 / Vienna 2023

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 821
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Mathematik UG2002
Betreut von / Supervisor:	Univ.-Prof. Dr. Radu Ioan Boţ

Abstract (English)

Finding the optimal value for a given problem is the goal in a variety of scientific fields. The mathematical field of optimization attempts to solve this problem under various assumptions. Because of their low computational cost and broad applicability, first-order methods are particularly popular. Gradient descent is a first-order method that works well for a variety of problems. Despite the fact that gradient descent has been known for a long time, it still performs well in comparison with newer approaches, especially for high-dimensional data. One issue that gradient descent faces is determining what stepsize to use. Many different approaches have been developed over the years, but they all had drawbacks, such as limited convergence guarantees or expensive subroutines to calculate the stepsize. Yura Malitsky and Konstantin Mishchenko took a novel approach in their paper [9], which was published in 2020. In this thesis, we will compare their new approach to other well-known gradient descent algorithms in terms of their theoretical approach to proving convergence as well as how well they perform on specific real-world machine learning problems.

Zusammenfassung (Deutsch)

Die Suche nach dem optimalen Wert für ein bestimmtes Problem ist das Ziel in einer Vielzahl von wissenschaftlichen Bereichen. Das mathematische Gebiet der Optimierung versucht, dieses Problem unter verschiedenen Annahmen zu lösen. Wegen ihrer geringen Rechenkosten und ihrer breiten Anwendbarkeit sind Methoden erster Ordnung besonders beliebt. Der Gradientenabstieg ist eine Methode erster Ordnung, die für eine Vielzahl von Problemen gut geeignet ist. Obwohl der Gradientenabstieg schon seit langem bekannt ist, schneidet er im Vergleich zu neueren Ansätzen immer noch gut ab, insbesondere bei hochdimensionalen Problemen. Ein Problem des Gradientenabstiegs ist die Bestimmung der zu verwendenden Schrittweite. Im Laufe der Jahre wurden viele verschiedene Ansätze entwickelt, die jedoch alle Nachteile hatten, wie z.B. begrenzte Konvergenzgarantien oder teure Unterprogramme zur Berechnung der Schrittweite. Yura Malitsky und Konstantin Mishchenko verfolgten in ihrer im Jahr 2020 veröffentlichten Arbeit [9] einen neuartigen Ansatz. In dieser Arbeit werden wir ihren neuen Ansatz mit anderen bekannten Gradientenabstiegsalgorithmen vergleichen, und zwar sowohl im Hinblick auf ihren theoretischen Ansatz zum Nachweis der Konvergenz als auch in Bezug auf ihre Leistung bei bestimmten realen Problemen des maschinellen Lernens.

Acknowledgements

First and foremost, I would like to express my sincere appreciation to my supervisor, Prof. Dr. Radu Ioan Boţ, for his continuous support while I worked on this thesis, as well as his fantastic lectures throughout my academic career.

I'd also like to thank my grandmother, Marianne Harrer, for her love and support from childhood to now, as well as my mother, Nicole Harrer, for her ongoing support.

Last but not least, I'd like to thank my dear friend Philip Neuhart, without whom I wouldn't have made it through the first semester.

Contents

1 Introduction

1.1 Unconstrained optimization

Unconstrained optimization deals with the following problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function.
We can define two types of solutions to this problem. For $\epsilon > 0$, we define the ϵ-ball $B_\epsilon(x^*) := \{x \in \mathbb{R}^n \mid \|x - x^*\| < \epsilon\}$, where $\|.\|$ denotes the Euclidean norm.

Definition 1.1.1. *We call x^* a local minimum of f if there exists an $\epsilon > 0$ s.t.*

$$f(x^*) \leqslant f(x) \quad \forall x \in B_\epsilon(x^*). \tag{1.2}$$

Definition 1.1.2. *We call x^* a global minimum of f if*

$$f(x^*) \leqslant f(x) \quad \forall x \in \mathbb{R}^n. \tag{1.3}$$

Remark 1.1.3. *We call x^* a strict-local minimum of f if (1.2) holds strictly $\forall x \in B_\epsilon(x^*) \backslash \{x^*\}$, and a strict-global minimum if (1.3) holds strictly $\forall x \in \mathbb{R}^n \backslash \{x^*\}$.*

1.2 Necessary optimality conditions

We need the following definition as well as the theorem after it in order to derive a condition that ensures optimality in regards to (1.2). We denote the standard scalar product of two vectors $x, y \in \mathbb{R}^n$ by $\langle x, y \rangle$.

Definition 1.2.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and $x \in \mathbb{R}^n$. Then $d \in \mathbb{R}^n$ is a called descent direction of f at x if*

$$\langle d, \nabla f(x) \rangle < 0.$$

A simple example for a descent direction is $d := -\nabla f(x)$, furthermore $d := -B\nabla f(x)$, for any positive definite matrix $B \in \mathbb{R}^{n \times n}$ is also a descent direction. The next statement can be found in [3, Theorem 2.11].

Theorem 1.2.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, $x \in \mathbb{R}^n$ such that $\nabla f(x) \neq 0$ and $d \in \mathbb{R}^n$ a descent direction of f at x. Then there exists $\eta > 0$ such that*

$$f(x + \alpha d) < f(x) \quad \forall 0 < \alpha \leqslant \eta. \tag{1.4}$$

Moreover, for any $\beta < 1$, there exists $\hat{\eta} > 0$ such that

$$f(x + \alpha d) < f(x) + \alpha\beta\langle\nabla f(x), d\rangle \tag{1.5}$$

for all $0 < \alpha \leqslant \widehat{\eta}$.

Theorem 1.2.3. *Let x^* be a local minimum of a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ in an open neighborhood $B_\epsilon(x^*)$. Then*

$$\nabla f(x^*) = 0. \tag{1.6}$$

If furthermore f is twice continuously differentiable on $B_\epsilon(x^*)$, then

$$\nabla^2 f(x^*) \text{ is positive semidefinite.} \tag{1.7}$$

Proof. Assuming $\nabla f(x^*) \neq 0$ implies that $\nabla f(x^*)$ is a descent direction and therefore there exists an $\eta > 0$ such that $f\left(x - \alpha\nabla f(x^*)\right) < f(x)\ \forall\alpha \leqslant \eta$, which is a contradiction to x^* beeing a local minimum. For the second statement assume that $\nabla^2 f(x^*)$ is not positive semidefinite. Therefore there exists $d \in \mathbb{R}^n$, such that $d^T\nabla^2 f(x^*)d < 0$, by continuity of $\nabla^2 f(x^*)$ there exists a $T > 0$ such that $d^T\nabla^2 f\left(x^* + td\right)d < 0$, holds $\forall t \in [0, T]$. Using Taylors theorem we get

$$f\left(x^* + \bar{t}d\right) = f\left(x^*\right) + \bar{t}d^T\nabla f\left(x^*\right) + \frac{1}{2}\bar{t}^2 d^T\nabla^2 f\left(x^* + \xi d\right)d \tag{1.8}$$

for some $\xi \in (0, \bar{t})$. Since $\nabla f(x^*) = 0$ and $d^T\nabla^2 f\left(x^* + \xi d\right)d < 0$, this is a contradiction to x^* being a local minimum. □

Remark 1.2.4. *Note that if f is twice differentiable, then, if $\nabla^2 f(x^*)$ is positive definite and* (1.6) *holds, x^* is a strict local minimum. For the proof see Theorem 5.7 in [3].*

Remark 1.2.5. *It is worth noting that if* (1.7) *holds for any x, it is equivalent to the function being convex if it is twice differentiable. For more information on convexity, see* (1.4.2).

Definition 1.2.6. *A point x^* is called a stationary point of f if $\nabla f(x^*) = 0$.*

1.3 Line search

Having derived an easy-to-check necessary condition for a local minimum, namely $\nabla f(x^*) = 0$, we will now use it to introduce an important class of algorithms to solve (1.1). This class of algorithms updates a current point x_k along a search direction d_k with a stepsize λ_k in the following way

$$x_{k+1} = x_k + \lambda_k d_k. \tag{1.9}$$

This class is known as line search strategy, and the choice $d_k := \nabla f(x_k)$ is called steepest descent.
In this following chapters we lay down the concepts needed to prove convergency for the algorithms proposed in section 1.7.

1.4 Convexity

An important concept is convexity.

Definition 1.4.1. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called convex if for all $x, y \in \mathbb{R}^n$*

$$f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda)f(y) \quad \forall \lambda \in [0,1], \tag{1.10}$$

holds.

See [10] for a proof of the following lemma.

Lemma 1.4.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and twice differentiable for 4. Then the following statements are equivalent.*

1. *f is convex,*
2. *$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geqslant 0 \quad \forall x, y \in \mathbb{R}^n$,*
3. *$f(y) \geqslant f(x) + \langle y - x, \nabla f(x) \rangle \quad \forall x, y \in \mathbb{R}^n$,*
4. *$\nabla^2 f(x)$ is positive semidefinite $\quad \forall x \in \mathbb{R}^n$.*

Remark 1.4.3. *Note that a local minimum of (1.1) is also a global minimum if f is convex.*

1.5 Strong convexity

To get stronger convergency results, we will also need the definition of strong convexity.

Definition 1.5.1. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called μ-strongly convex with modules $\mu > 0$, if for all $x, y \in \mathbb{R}^n$*

$$f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda)f(y) - \frac{\mu}{2}\lambda(1-\lambda)\|x-y\|_2^2 \quad \forall \lambda \in [0,1], \tag{1.11}$$

holds.

Lemma 1.5.2. *The following conditions are equivalent to the fact that f is μ-strongly convex*

1. *$g(x) := f(x) - \frac{\mu}{2}\|x\|^2$ is convex, $\forall x \in \mathbb{R}^n$.*
2. *$(\nabla f(x) - \nabla f(y))^T (x - y) \geqslant \mu \|x - y\|^2, \forall x, y \in \mathbb{R}^n$.*
3. *$f(y) \geqslant f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2}\|y - x\|^2, \forall x, y \in \mathbb{R}^n$.*

Proof. These follow from the different statements in (1.4.2) applied to g. □

Remark 1.5.3. *Note that the minimization problem (1.1) has a unique solution if f is strongly convex.*

1.6 L-smoothness

An important concept for many convergence proofs is L-smoothness.

Definition 1.6.1. *A differential function $f : \mathbb{R}^n \to \mathbb{R}$ is called L-smooth if there exists $L \geqslant 0$ such that*

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\| \quad \forall x, y \in \mathbb{R}^n. \tag{1.12}$$

In order to derive a result about convergence, we will need the following lemmata, which are taken from [12].

Lemma 1.6.2. *If the function $f : \mathbb{R}^n \to \mathbb{R}$ is differential and L-smooth, then it holds*

$$|f(y) - f(x) - \langle \nabla f(x), y - x\rangle| \leqslant \frac{L}{2}\|y - x\|^2 \quad \forall x, y \in \mathbb{R}^n. \tag{1.13}$$

Proof. Let $x, y \in \mathbb{R}^n$ and define $g(\tau) := f(x + \tau(y - x))$, for $\tau \in [0, 1]$,
which implies $g'(\tau) = \langle \nabla f(x + \tau(y - x), y - x\rangle$.
By the fundamental theorem of calculus we have that

$$\int_0^1 \langle \nabla f(x + \tau(y - x)), y - x\rangle d\tau = \int_0^1 g'(t)d\tau = g(1) - g(0) = f(y) - f(x),$$

it follows that

$$f(y) = f(x) + \langle \nabla f(x), y - x\rangle + \int_0^1 \langle \nabla f(x + \tau(y - x)) - \nabla f(x), y - x\rangle d\tau.$$

Thus

$$\begin{aligned}
&|f(y) - f(x) - \langle \nabla f(x), y - x\rangle| \\
&= \left| \int_0^1 \langle \nabla f(x + \tau(y - x)) - \nabla f(x), y - x\rangle d\tau \right| \\
&\leqslant \int_0^1 \|\nabla f(x + \tau(y - x)) - \nabla f(x)\| \cdot \|y - x\| d\tau \\
&\leqslant \int_0^1 \tau L\|y - x\|^2 d\tau = \frac{L}{2}\|y - x\|^2.
\end{aligned}$$

□

Geometrically, this means that f is bounded by two quadratic functions in the following way. Let

$$\begin{aligned}
\phi_1(x) &= f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2, \\
\phi_2(x) &= f(y) + \langle \nabla f(y), x - y\rangle - \frac{L}{2}\|x - y\|^2,
\end{aligned}$$

then

$$\phi_1(x) \geqslant f(x) \geqslant \phi_2(x) \quad \forall x \in \mathbb{R}^n.$$

We will also need the following fact later on.

Lemma 1.6.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex, differentiable and L-smooth function. Then*

$$f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2L} \|\nabla f(x) - \nabla f(y)\|^2 \leqslant f(y) \quad \forall x, y \in \mathbb{R}^n. \tag{1.14}$$

Proof. Let us fix $x \in \mathbb{R}^n$ and define,

$$\phi(y) := f(y) - \langle \nabla f(x), y \rangle .$$

Note that ϕ is convex and L-smooth and its optimal point is $y^* = x$, since $\nabla\phi(y) = \nabla f(y) - \nabla f(x)$ and the function is convex. We see that by using L-smoothness, in particular (1.13) and Cauchy–Schwarz inequality on ϕ,

$$\phi(y^*) \leqslant \phi\left(y - \frac{1}{L}\nabla\phi(y)\right) \leqslant \phi(y) - \frac{1}{2L}\|\nabla\phi(y)\|^2 .$$

Therefore, we get

$$\phi(x) \leqslant \phi(y) - \frac{1}{2L}\|\nabla\phi(y)\|^2,$$

which yields the result. □

1.7 Line search routines

In this section, we will list common line search techniques, in order to compare them with the novel strategy proposed by Yura Malitsky and Konstantin Mishchenko [9].
The main challenge we face is the determination of the optimal stepsize. One simple way to do so is to choose a constant stepsize, i.e., $\lambda_k = \lambda$ for all k.
Another way to solve this problem is the following algorithm, known as "exact line search".

Algorithm 1.7.1: Exact line search

Input: A sequence of descent directions $(d_k)_{k\in\mathbb{N}}$, a starting point x_0 and a stopping criterion.

Routine:

Set $k := 0$
while *stopping criterion is not satisfied* **do**
  $\lambda_k := \min_{\lambda \geqslant 0} f(x_k + \lambda d_k)$
  $x_{k+1} = x_k - \lambda_k d_k$
  $k = k + 1$

A common stopping criterion is $\|\nabla f(x_k)\| < \epsilon$, for a small and fixed $\epsilon > 0$. This approach faces two major problems. First of all, even though it reduces the original problem to a one-dimensional problem, this new one-dimensional problem can be very hard to solve. Secondly, without preconditioning, the convergence to a solution is very slow due to zigzagging as the next theorem shows.
In the following we assume that we have no stopping criterion, and therefore algorithm (1.7.1) generates an infinite sequence of iterates $(x_k)_{k\in\mathbb{N}}$.

Theorem 1.7.2. *Let $(x_k)_{k\in\mathbb{N}}$ be the sequence generated by the exact line search algorithm with $d := \nabla f(x_k)$. Then*

$$\langle x_{k+1} - x_k, x_{k+2} - x_{k+1}\rangle = 0 \text{ for any } k \in \mathbb{N}. \tag{1.15}$$

Proof. By definition we have that

$$\langle x_{k+1} - x_k, x_{k+2} - x_{k+1}\rangle = \lambda_k\lambda_{k+1}\langle \nabla f(x_k), \nabla f(x_{k+1})\rangle,$$

and λ_k minimizes $\varphi_k(\lambda) := f(x_k - \lambda\nabla f(x_k))$, which implies $\frac{d\varphi_k}{d\lambda}(\lambda_k) = 0$. This concludes the proof since

$$\frac{d\varphi_k}{d\lambda}(\lambda_k) = \langle -\nabla f(x_k), \nabla f(x_k - \lambda_k\nabla f(x_k))\rangle = \langle -\nabla f(x_k), \nabla f(x_{k+1})\rangle.$$

□

Rather than determining the minimal value for the step size, we can come up with other routines for the step size. Such a method is called inexact line search. One example for this is to choose $\lambda_k = \frac{g_k^T g_k}{g_k^T G g_k}$, where $g_k := \nabla f(x_k)$ and G denotes the Hessian matrix of f. This step coincides with the exact line search if a quadratic model is chosen.

Algorithm 1.7.3: Constant stepsize

Input: A sequence of descent directions $(d_k)_{k\in\mathbb{N}}$, a starting point x_0, a stepsize $\lambda > 0$ and a stopping criterion.

Routine:

```
Set k := 0
while a stopping criterion is not satisfied do
    λ_k := λ
    x_{k+1} = x_k − λ_k d_k
    k = k + 1
```

Wolfe conditions

To ensure that computing the stepsize takes as little time as possible, we need to find a condition that can guarantee convergence of our algorithm if it is satisfied and also computationally inexpensive to verify. The condition that the algorithm is a descent method i.e. $f(x_k + \lambda_k d_k) < f(x_k)$ is not enough, see Example 11.5 in [3]. In this example, the step size is chosen so large that the fact that we are going along a descent direction doesn't matter anymore. The fact that we managed to produce a descending sequence is pure luck. Longer steps should only be accepted if they also result in greater improvements than shorter steps, or more precisely, the decrease should be proportional to the step size.

The following definition captures this concept.

Definition 1.7.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, d_k a descent direction of f at x_k and a stepsize $\lambda_k > 0$. We say that the function f decreases sufficiently along λ_k, if there exists a $\beta_1 \in (0,1)$ such that*

$$f\left(x_k + \lambda_k d_k\right) \leqslant f\left(x_k\right) + \lambda_k \beta_1 \langle \nabla f\left(x_k\right), d_k \rangle.$$

This condition is known as the first Wolfe condition or the Armijo condition, see [18] and [2]. By (1.5) such a stepsize always exists.
This condition gives rise to the inexact line search method described bellow.

Algorithm 1.7.5: Armijo line search

Input: $s > 0, \beta_1 \in (0,1)$ and $\sigma_1 \in (0,1)$.
Routine:
1 Set $m := 0$
2 **while** $f(x_k + \sigma_1^m s d_k) > f(x_k) + \sigma_1^m \beta_1 \langle \nabla f(x_k), d_k \rangle$ **do**
3 $\quad m = m + 1$
`/* decrease stepsize until the first Wolfe condition is satisfied */`
4 $\lambda_k := \sigma_1^m s$

To prevent too-small step sizes, we introduce the following definition.

Definition 1.7.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function, d_k a descent direction in x_k and a stepsize $\lambda_k > 0$. We say that the step made sufficient progress compared to x_k, if there exists a $\beta_2 \in (0,1)$ such that*

$$\langle \nabla f\left(x_k + \lambda_k d_k\right), d_k \rangle \geqslant \beta_2 \langle \nabla f\left(x_k\right), d_k \rangle.$$

This is known as the second Wolfe condition. The two Wolfe conditions give rise to the following algorithm attributed too Fletcher [6] and Lemaréchal [8].

Algorithm 1.7.7: Line search

Input: $\alpha_0 > 0, \beta \in (0,1), \sigma_1 \in (0,1), \sigma_2 \in (0,1)$ with $\sigma_1 < \sigma_2$.

Routine:

$i := 0$
$\alpha_\ell := 0$
$\alpha_r := +\infty$
repeat
  if $f(x_k + \alpha_i d_k) > f(x_k) + \alpha_i \langle \nabla f(x_k), d_k \rangle$ **then**
    $\alpha_r = \alpha_i$
    $\alpha_{i+1} = \frac{\alpha_\ell + \alpha_r}{2}$.
  `/* decrease stepsize if first Wolfe condition is not satisfied */`
  else if $\langle \nabla f(x_k + \alpha_k d_k), d_k \rangle < \beta_2 \langle \nabla f(x_k), d_k \rangle$ **then**
    $\alpha_\ell := \alpha_i$
    if $\alpha_r < +\infty$ **then**
      $\alpha_{i+1} := \frac{\alpha_\ell + \alpha_r}{2}$
    else
      $\alpha_{i+1} := \lambda \alpha_i$
  `/* increase stepsize if second Wolfe condition is not satisfied */`
until α_j *satisfies both Wolfe conditions.*
$\lambda_k := \alpha_j$

Remark 1.7.8. *Since* $\langle \nabla f(x_k + \lambda^* d_k), d_k \rangle = 0$, *for* $\lambda^* := \min_{\lambda \geqslant 0} f(x_k + \lambda d_k)$ *and* $\langle \nabla f(x_k), d_k \rangle < 0$, *by the mean value theorem there must exist a stepsize such that the second Wolfe condition is satisfied.*

For the proof of the following statement see Theorem 11.9 in [3].

Lemma 1.7.9. *If* $0 < \beta_1 < \beta_2 < 1$, *then there exists a stepsize* $\lambda > 0$ *such that the first and second Wolfe condition are both satisfied.*

Newton's method

We recall Newton's method in order to derive another update rule which follows right after this. Using the fact that $\nabla f(x^*)$ has to be zero at a local minimum as the first order condition (1.6) stated. We can try to solve this equation to find candidates for the minimization problem (1.1).

Using Newton's method to solve this system of equations is one way to do so.

Algorithm 1.7.10: Newton's local method

Input: $x_0 \in \mathbb{R}^n$.

Routine:

Set $k := 0$

while *stopping criterion is not satisfied* **do**

  Calculate d_k as a solution of $\nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$

  $x_{k+1} := x_k + d_k$

  $k := k + 1$

Under the right conditions, this algorithm converges very quickly. The obvious drawbacks of this algorithm are that the calculation of the Hessian matrix can be expensive, and furthermore, the starting point has to be close to a solution, otherwise the algorithm does not have to converge.

Quasi-Newton methods

Quasi-Newton methods use an approximation of the Hessian matrix in order to be computationally feasible compared to the regular Newton method, see section 5.1.1 in [16] for more information. One example for this is the Barzilian-Borwein stepsize, in which λ_k has the form $\lambda_k = \alpha_k$ such that α_k minimizes $\|\Delta x - \alpha \Delta g\|^2$, with $\Delta x = x_k - x_{k-1}$ and $\Delta g = \nabla f(x_k) - \nabla f(x_{k-1})$. This gives an approximation to the quasi-Newton equation. Solving this least-squares problem yields

$$\lambda_k = \frac{\langle \Delta x, \Delta g \rangle}{\langle \Delta g, \Delta g \rangle} = \frac{\langle x_k - x_{k-1}, \nabla f(x_k) - \nabla f(x_{k-1}) \rangle}{\langle \nabla f(x_k) - \nabla f(x_{k-1}), \nabla f(x_k) - \nabla f(x_{k-1}) \rangle}. \tag{1.16}$$

Alternatively one can also minimize $\|\alpha \Delta x - \Delta g\|^2$, which yields

$$\lambda_k = \frac{\langle \Delta x, \Delta x \rangle}{\langle \Delta x, \Delta g \rangle} = \frac{\langle x_k - x_{k-1}, x_k - x_{k-1} \rangle}{\langle x_k - x_{k-1}, \nabla f(x_k) - \nabla f(x_{k-1}) \rangle}. \tag{1.17}$$

These two stepsizes are often regarded as BB1 and BB2 stepsizes. Other famous Quasi-Newton methods are the BFGS update and SR1 update.

In 1983, Nesterov [11] introduced the following idea for convex and L-smooth problems.

Nesterov Accelerated Gradient

A more sophisticated update rule in each step is Nesterov's approach [11].

Algorithm 1.7.11: Nesterov Accelerated Gradient method

Input: $x_0 = x_1 \in \mathbb{R}^n, \lambda \in \left(0, \frac{1}{L}\right]$.
Routine:
Set $k := 1$, $t_1 := 1$, $x_1 := x_0$
while *stopping criterion is not satisfied* **do**
  $t_{k+1} := \frac{1+\sqrt{4t_k^2+1}}{2}$
  $y^k := x^k + \frac{t_k - 1}{t_{k+1}}\left(x^k - x^{k-1}\right)$
  $x^{k+1} := y^k - \lambda \nabla f\left(y^k\right)$
  $k := k + 1$

The sequence $\left(y^k\right)_{k \geqslant 1}$ is called the momentum sequence.

1.8 Convergence results

We will now state convergence results for the constant stepsize version of gradient descent algorithm 1.7.3. We will need the following definition.

Definition 1.8.1. *A sequence $(x_k)_{k\in\mathbb{N}}$ is called linearly convergent to x^* if there exists a constant $c \in (0,1)$ and $N \in \mathbb{N}$ such that $\|x_{i+1} - x^*\| \leqslant c\|x_i - x^*\|$ for all $i \geqslant N$.*

Theorem 1.8.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-smooth and bounded from bellow. If the fixed step size satisfies $0 < \lambda < \frac{2}{L}$, then the gradient descent algorithm 1.7.3 without a stopping criterion will generate a sequence $(x_k)_{k\in\mathbb{N}}$ such that $\lim_{k\to\infty} \nabla f(x_k) = 0$, for any starting point x_0.*

Proof. Let $k \in \mathbb{N}$ be fixed, by (1.13) and the definition of x_{k+1}, we have

$$\begin{aligned} f(x_{k+1}) &\leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &= f(x_k) - \lambda \|\nabla f(x_k)\|^2 + \frac{\lambda^2 L}{2}\|\nabla f(x_k)\|^2 \\ &= f(x_k) - \lambda\left(1 - \frac{\lambda L}{2}\right)\|\nabla f(x_k)\|^2, \end{aligned} \tag{1.18}$$

which implies,

$$\|\nabla f(x_k)\|^2 \leqslant \frac{1}{\lambda\left(1 - \frac{\lambda L}{2}\right)}\left(f(x_k) - f(x_{k+1})\right). \tag{1.19}$$

Let $\underline{f}$ be the lower bound of f, and fix $N \in \mathbb{N}$

$$\begin{aligned}\sum_{k=0}^{N} \|\nabla f(x_k)\|^2 &\leqslant \frac{1}{\lambda\left(1-\frac{\lambda}{2}L\right)} \sum_{k=0}^{N} f(x_k) - f(x_{k+1}) \\ &\leqslant \frac{1}{\lambda\left(1-\frac{\lambda}{2}L\right)} \big(f(x_0) - f(x_{N+1})\big). \\ &\leqslant \frac{1}{\lambda\left(1-\frac{\lambda}{2}L\right)} \big(f(x_0) - \underline{f}\big),\end{aligned} \tag{1.20}$$

which implies $\sum_{k=0}^{\infty} \|\nabla f(x_k)\|^2 < \infty$, and therefore $\lim_{k\to\infty} \nabla f(x_k) = 0$. □

Theorem 1.8.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and L-smooth with constant $L > 0$. Denote x^* as any solution of the minimization problem (1.1). Then with a fixed stepsize $\lambda \leqslant \frac{1}{L}$, the gradient descent algorithm 1.7.3 without a stopping criterion will produce a sequence $(x_k)_{k\in\mathbb{N}}$ such that*

$$f(x_N) - f(x^*) \leqslant \frac{\|x_0 - x^*\|_2^2}{2\lambda N} \quad \forall N \geqslant 1. \tag{1.21}$$

Proof. Let $k \in \mathbb{N}$ be fixed, by L-smoothness and the definition of x_{k+1}, we have again like in (1.18)

$$f(x_{k+1}) \leqslant f(x_k) - \lambda\left(1 - \frac{\lambda L}{2}\right)\|\nabla f(x_k)\|^2. \tag{1.22}$$

Since $1 - \frac{\lambda L}{2} \leqslant \frac{1}{2}$ by $\lambda \leqslant \frac{1}{L}$, we have

$$f(x_{k+1}) \leqslant f(x_k) - \lambda\frac{1}{2}\|\nabla f(x_k)\|_2^2. \tag{1.23}$$

By convexity, we have

$$f(x_k) \leqslant f(x^*) + \langle \nabla f(x_k), x_k - x^* \rangle, \tag{1.24}$$

which implies

$$\begin{aligned} f(x_{k+1}) &\leqslant f(x^*) + \langle \nabla f(x_k), x_k - x^* \rangle - \frac{\lambda}{2}\|\nabla f(x_k)\|_2^2 \\ f(x_{k+1}) - f(x^*) &\leqslant \frac{1}{2\lambda}\left(2\lambda\langle \nabla f(x_k), x_k - x^* \rangle - \lambda^2\|\nabla f(x_k)\|_2^2\right), \end{aligned} \tag{1.25}$$

and therefore

$$\begin{aligned} f(x_{k+1}) - f(x^*) &\leqslant \frac{1}{2\lambda}\left(2\lambda\langle \nabla f(x_k), x_k - x^* \rangle - \lambda^2\|\nabla f(x_k)\|_2^2 - \|x_k - x^*\|_2^2 + \|x_k - x^*\|_2^2\right) \\ &= \frac{1}{2\lambda}\left(\|x_k - x^*\|_2^2 - \|x_k - \lambda\nabla f(x_k) - x^*\|_2^2\right). \end{aligned} \tag{1.26}$$

Therefore, we have

$$f\left(x_{k+1}\right)-f\left(x^*\right) \leqslant \frac{1}{2\lambda}\left(\left\|x_k-x^*\right\|_2^2-\left\|x_{k+1}-x^*\right\|_2^2\right). \tag{1.27}$$

Letting $N>1$ and summing up this inequality yields

$$\begin{aligned}\sum_{k=1}^{N}\left(f\left(x_k\right)-f\left(x^*\right)\right) &\leqslant \sum_{k=1}^{N}\frac{1}{2\lambda}\left(\left\|x_{k-1}-x^*\right\|_2^2-\left\|x_k-x^*\right\|_2^2\right)\\ &=\frac{1}{2\lambda}\left(\left\|x_0-x^*\right\|_2^2-\left\|x_N-x^*\right\|_2^2\right)\\ &\leqslant \frac{1}{2\lambda}\left(\left\|x_0-x^*\right\|_2^2\right).\end{aligned} \tag{1.28}$$

(1.23) implies that $f(x_{k+1}) \leqslant f(x_k)$ for any k and therefore, we get

$$\begin{aligned}f\left(x_N\right)-f\left(x^*\right) &\leqslant \frac{1}{N}\sum_{k=1}^{N} f(x_k)-f(x^*)\\ &\leqslant \frac{\left\|x_0-x^*\right\|_2^2}{2\lambda N}.\end{aligned} \tag{1.29}$$

□

Theorem 1.8.4. *Let $f:\mathbb{R}^n\to\mathbb{R}$ be L-smooth and μ-strongly convex. Denote x^* as the solution of the minimization problem* (1.1)*. From a given $x_0\in\mathbb{R}^n$ and $\lambda\leqslant\frac{1}{L}$, the gradient descent algorithm with constant stepsize* (1.7.3) *converges according to*

$$\left\|x_{k+1}-x^*\right\|_2^2 \leqslant (1-\lambda\mu)^{k+1}\left\|x_0-x^*\right\|_2^2 \quad \forall k\geqslant 0.$$

For $\lambda=\frac{1}{L}$ this means linear convergence with the rate $\frac{\mu}{L}$.

Proof. Let $k\geqslant 1$. Denoting $a=x_{k+1}-x_k$ and $b=x_k-x^*$ we see that

$$\begin{aligned}\left\|x_{k+1}-x^*\right\|^2 &= \|a+b\|^2=\langle a,a\rangle+\langle a,b\rangle+\langle b,a\rangle+\langle b,b\rangle\\ &=\left\|x_k-x^*\right\|^2+2\langle x_{k+1}-x_k,x_k-x^*\rangle+\left\|x_{k+1}-x_k\right\|^2\\ &=\left\|x_k-x^*\right\|^2+2\lambda\langle\nabla f(x_k),x^*-x_k\rangle+\left\|x_{k+1}-x_k\right\|^2\\ &=\left\|x_k-x^*\right\|_2^2-2\lambda\langle\nabla f(x_k),x_k-x^*\rangle+\lambda^2\|\nabla f(x_k)\|_2^2.\end{aligned} \tag{1.30}$$

Strong convexity followed by L-smoothness (1.14) implies

$$\begin{aligned}&\left\|x_k-x^*\right\|_2^2-2\lambda\langle\nabla f(x_k),x_k-x^*\rangle+\lambda^2\|\nabla f(x_k)\|_2^2\\ &\qquad\leqslant(1-\lambda\mu)\left\|x_k-x^*\right\|_2^2-2\lambda\left(f(x_k)-f(x^*)\right)+\lambda^2\|\nabla f(x_k)\|_2^2\\ &\qquad\leqslant(1-\lambda\mu)\left\|x_k-x^*\right\|_2^2-2\lambda\left(f(x_k)-f(x^*)\right)+2\lambda^2L\left(f(x_k)-f(x^*)\right)\\ &\qquad=(1-\lambda\mu)\left\|x_k-x^*\right\|_2^2-2\lambda(1-\lambda L)\left(f(x_k)-f\left(x^*\right)\right).\end{aligned} \tag{1.31}$$

Since $\frac{1}{L} \geqslant \lambda$ we have $-2\lambda(1-\lambda L) \leqslant 0$, and therefore

$$\|x_{k+1} - x^*\|_2^2 \leqslant (1-\lambda\mu)\|x_k - x^*\|_2^2, \tag{1.32}$$

which yields the result. □

Remark 1.8.5. *The complexity to get $\|x^k - x^*\|^2 \leqslant \varepsilon$ is $\mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$, where $\kappa = \frac{L}{\mu}$.*

Nesterovs approach is under the right assumptions faster as the next theorem taken from [1] shows.

Theorem 1.8.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-smooth and convex, then the sequences $(f(x_k))_{k\in\mathbb{N}}$ produced by Nesterov's approach (1.7.11) converges to the optimal value f^* as*

$$f(x_k) - f^* \leqslant \frac{2L\|x_0 - x^*\|_2^2}{k^2} \quad \forall k \geqslant 1. \tag{1.33}$$

Remark 1.8.7. *It is not known if $(x_k)_{k\in\mathbb{N}}$ converges for Nesterov's approach.*

2 An adaptive stepsize based on the local smoothness constant

Having seen a few common methods, we will now focus on the newly developed idea by Yura Malitsky and Konstantin Mishchenko, [9].

2.1 Adaptive gradient descent

The following algorithm can be motivated by the idea that the stepsize should be a local approximation to $\frac{1}{L}$, where L denotes the L-smoothness constant.

Algorithm 2.1.1: Adaptive Gradient Descent

Input: A starting point $x_0 \in \mathbb{R}^n$, an initial first stepsize $\lambda_0 > 0$ and $\theta_0 = +\infty$

Routine:

$k := 1$

$x_1 = x_0 - \lambda_0 \nabla f(x_0)$

while *stopping criterion is not satisfied* **do**

  $\lambda_k = \min\left\{\sqrt{1+\theta_{k-1}}\lambda_{k-1}, \frac{\|x_k - x_{k-1}\|}{2\|\nabla f(x_k) - \nabla f(x_{k-1})\|}\right\}$

  $x_{k+1} = x_k - \lambda_k \nabla f(x_k)$

  $\theta_k = \frac{\lambda_k}{\lambda_{k-1}}$

  $k = k + 1$

The definition of λ_k implies for all $k \geqslant 1$ the following two inequalities,

$$\lambda_k^2 \leqslant (1 + \theta_{k-1})\,\lambda_{k-1}^2 \tag{2.1}$$

$$\lambda_k \leqslant \frac{\|x_k - x_{k-1}\|}{2\,\|\nabla f(x_k) - \nabla f(x_{k-1})\|} \tag{2.2}$$

It will be clear from (2.1.5) that inequality (2.1) is required to prove convergency.

Preparation

We need the following lemma later on.

Lemma 2.1.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and differentiable. Denote x^* as any solution of the minimization problem (1.1) and f_* as the minimal value. Then the following inequality holds for $(x_k)_{k\in\mathbb{N}}$ generated by algorithm 2.1.1.*

$$\begin{aligned}\|x_{k+1} - x^*\|^2 + \frac{1}{2}\|x_{k+1} - x_k\|^2 + 2\lambda_k(1+\theta_k)(f(x_k) - f_*) \\ \leqslant \|x_k - x^*\|^2 + \frac{1}{2}\|x_k - x_{k-1}\|^2 + 2\lambda_k\theta_k(f(x_{k-1}) - f_*) \quad \forall k \geqslant 1.\end{aligned} \tag{2.3}$$

Proof. Let $k \geqslant 1$. Again denoting $a = x_{k+1} - x_k$ and $b = x_k - x^*$ we see that

$$\begin{aligned}\|x_{k+1} - x^*\|^2 &= \|a + b\|^2 = \langle a, a\rangle + \langle a, b\rangle + \langle b, a\rangle + \langle b, b\rangle \\ &= \|x_k - x^*\|^2 + 2\langle x_{k+1} - x_k, x_k - x^*\rangle + \|x_{k+1} - x_k\|^2 \\ &= \|x_k - x^*\|^2 + 2\lambda_k\langle \nabla f(x_k), x^* - x_k\rangle + \|x_{k+1} - x_k\|^2.\end{aligned} \tag{2.4}$$

By convexity of f we get

$$f(y) \geqslant f(x) + \langle \nabla f(x), y - x\rangle \text{ for all } x, y \in \mathbb{R}^n. \tag{2.5}$$

Which for $x = x_k$ and $y = x^*$ yields,

$$2\lambda_k\langle \nabla f(x_k), x^* - x_k\rangle \leqslant 2\lambda_k(f_* - f(x_k)), \tag{2.6}$$

and therefore

$$\|x_{k+1} - x^*\|^2 \leqslant \|x_k - x^*\|^2 - 2\lambda_k(f(x_k) - f_*) + \|x_{k+1} - x_k\|^2. \tag{2.7}$$

We now look at the following term:

$$\begin{aligned}\|x_{k+1} - x_k\|^2 &= 2\|x_{k+1} - x_k\|^2 - \|x_{k+1} - x_k\|^2 \\ &= -2\lambda_k\langle \nabla f(x_k), x_{k+1} - x_k\rangle - \|x_{k+1} - x_k\|^2 \\ &= 2\lambda_k\langle \nabla f(x_k) - \nabla f(x_{k-1}), x_k - x_{k+1}\rangle \\ &\qquad + 2\lambda_k\langle \nabla f(x_{k-1}), x_k - x_{k+1}\rangle - \|x_{k+1} - x_k\|^2\end{aligned} \tag{2.8}$$

Now we estimate the first two terms on the right-hand side of (2.8). Using the definition of λ_k, followed by Cauchy–Schwarz inequality as well as Young's inequality for products, yields

$$\begin{aligned}2\lambda_k\langle \nabla f(x_k) - \nabla f(x_{k-1}), x_k - x_{k+1}\rangle &\leqslant 2\lambda_k\|\nabla f(x_k) - \nabla f(x_{k-1})\|\|x_k - x_{k+1}\| \\ &\leqslant \|x_k - x_{k-1}\|\|x_k - x_{k+1}\| \\ &\leqslant \frac{1}{2}\|x_k - x_{k-1}\|^2 + \frac{1}{2}\|x_{k+1} - x_k\|^2.\end{aligned} \tag{2.9}$$

Secondly, using convexity of f, again similar as in (2.6), we see that

$$\begin{aligned} 2\lambda_k\langle\nabla f(x_{k-1}), x_k - x_{k+1}\rangle = \frac{2\lambda_k}{\lambda_{k-1}}\langle x_{k-1} - x_k, x_k - x_{k+1}\rangle = 2\lambda_k\theta_k\langle x_{k-1} - x_k, \nabla f(x_k)\rangle \\ \leqslant 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)). \end{aligned} \tag{2.10}$$

Plugging (2.9) and (2.10) into (2.8), we obtain

$$\|x_{k+1} - x_k\|^2 \leqslant \frac{1}{2}\|x_k - x_{k-1}\|^2 - \frac{1}{2}\|x_{k+1} - x_k\|^2 + 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)). \tag{2.11}$$

Using (2.11) in (2.7), we get the following inequality

$$\begin{aligned} \|x_{k+1} - x^*\|^2 \leqslant \|x_k - x^*\|^2 - 2\lambda_k(f(x_k) - f_*) + \frac{1}{2}\|x_k - x_{k-1}\|^2 \\ - \frac{1}{2}\|x_{k+1} - x_k\|^2 + 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)). \end{aligned} \tag{2.12}$$

Rearranging the equation yields the following

$$\begin{aligned} \|x_{k+1} - x^*\|^2 + \frac{1}{2}\|x_{k+1} - x_k\|^2 + 2\lambda_k(f(x_k) - f_*) \\ \leqslant \|x_k - x^*\|^2 + \frac{1}{2}\|x_k - x_{k-1}\|^2 + 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)), \end{aligned} \tag{2.13}$$

after adding and subtraction $2\lambda_k\theta_k f_*$ this gives us

$$\begin{aligned} \|x_{k+1} - x^*\|^2 + \frac{1}{2}\|x_{k+1} - x_k\|^2 + 2\lambda_k(f(x_k) - f_*) \\ \leqslant \|x_k - x^*\|^2 + \frac{1}{2}\|x_k - x_{k-1}\|^2 + 2\lambda_k\theta_k f(x_{k-1}) - 2\lambda_k\theta_k(f(x_k) - f_*) - 2\lambda_k\theta_k f_*, \end{aligned} \tag{2.14}$$

which, after another rearrangement, is exactly (2.3). □

In order to prove convergence, we will need the following lemma.

Lemma 2.1.3. *Let $(x_k)_{k\in\mathbb{N}}$ be a sequence in $\mathbb{R}^n$ and $(a_k)_{k\in\mathbb{N}}$ in $\mathbb{R}_+$. Suppose that $(x_k)_{k\in\mathbb{N}}$ is bounded, its cluster points belong to $\mathcal{X} \subset \mathbb{R}^n$, and it also holds that*

$$\|x_{k+1} - x\|^2 + a_{k+1} \leqslant \|x_k - x\|^2 + a_k, \quad \forall x \in \mathcal{X}. \tag{2.15}$$

Then $\lim_{k\to\infty} x_k \in \mathcal{X}$ exists.

Proof. For any two cluster points $\bar{x}_1, \bar{x}_2$, there exist two subsequences $(x_{k_i})_{i\in\mathbb{N}}$ and $(x_{k_j})_{j\in\mathbb{N}}$ such that $x_{k_i} \to \bar{x}_1$ and $x_{k_j} \to \bar{x}_2$. Since $\left(\|x_k - x\|^2 + a_k\right)_{k\in\mathbb{N}}$ is nonnegative and decreasing, $\lim_{k\to\infty}\left(\|x_k - x\|^2 + a_k\right)$ exists for any $x \in \mathcal{X}$. Let $x = \bar{x}_1$, we observe

$$\lim_{k\to\infty}\left(\|x_k - \bar{x}_1\|^2 + a_k\right) = \lim_{i\to\infty}\left(\|x_{k_i} - \bar{x}_1\|^2 + a_{k_i}\right) = \lim_{i\to\infty} a_{k_i}, \tag{2.16}$$

and also

$$\lim_{k\to\infty}\left(\|x_k-\bar{x}_1\|^2+a_k\right)=\lim_{j\to\infty}\left(\|x_{k_j}-\bar{x}_1\|^2+a_{k_j}\right)=\|\bar{x}_2-\bar{x}_1\|^2+\lim_{j\to\infty}a_{k_j}. \tag{2.17}$$

Therefore $\lim_{i\to\infty}a_{k_i}$ exists and $\lim_{i\to\infty}a_{k_i}=\lim_{j\to\infty}a_{k_j}+\|\bar{x}_1-\bar{x}_2\|^2$ holds. Doing the same for $x=\bar{x}_2$ yields that $\lim_{i\to\infty}a_{k_i}$ exists and $\lim_{j\to\infty}a_{k_j}=\lim_{i\to\infty}a_{k_i}+\|\bar{x}_1-\bar{x}_2\|^2$ holds. As a result, we get $\bar{x}_1=\bar{x}_2$, and since this holds for arbitrary cluster points, we get the result.

□

Convergence to minimal value

Definition 2.1.4. *A function $f:\mathbb{R}^n\to\mathbb{R}$ is called locally Lipschitz if it is Lipschitz over any compact set of its domain. A function f with locally Lipschitz gradient ∇f is called locally smooth.*

Theorem 2.1.5. *Suppose that $f:\mathbb{R}^n\to\mathbb{R}$ is a convex and locally L-smooth function. Let f_* denote the minimal value to the minimization problem* (1.1)*. Then $(x_k)_{k\in\mathbb{N}}$ generated by algorithm 2.1.1 converges to a solution x^* and*

$$f(\hat{x}_k)-f_*\leqslant\frac{D}{2S_k}=\mathcal{O}\left(\frac{1}{k}\right)\quad\forall k\geqslant 1, \tag{2.18}$$

where

$$\begin{aligned}
D&=\|x_1-x^*\|^2+\frac{1}{2}\|x_1-x_0\|^2+2\lambda_1\theta_1\left[f\left(x_0\right)-f_*\right],\\
\hat{x}_k&=\frac{\lambda_k\left(1+\theta_k\right)x_k+\sum_{i=1}^{k-1}w_ix_i}{S_k},\\
w_i&=\lambda_i\left(1+\theta_i\right)-\lambda_{i+1}\theta_{i+1},\\
S_k&=\lambda_k\left(1+\theta_k\right)+\sum_{i=1}^{k-1}w_i=\sum_{i=1}^{k}\lambda_i+\lambda_1\theta_1.
\end{aligned} \tag{2.19}$$

Proof. Let $k\geqslant 1$. We first prove convergence of $(x_k)_{k\in\mathbb{N}}$. Telescoping (2.3), we deduce

$$\begin{aligned}
&\|x_{k+1}-x^*\|^2+\frac{1}{2}\|x_{k+1}-x_k\|^2+2\lambda_k\left(1+\theta_k\right)\left(f(x_k)-f_*\right)\\
&+2\sum_{i=1}^{k-1}\left[\lambda_i\left(1+\theta_i\right)-\lambda_{i+1}\theta_{i+1}\right]\left(f\left(x_i\right)-f_*\right)\\
&\leqslant\|x_1-x^*\|^2+\frac{1}{2}\|x_1-x_0\|^2+2\lambda_1\theta_1\left[f\left(x_0\right)-f_*\right]=D.
\end{aligned} \tag{2.20}$$

Note that inequality (2.1) is equivalent to $\lambda_k\theta_k\leqslant(1+\theta_{k-1})\lambda_{k-1}$, and therefore the second line above is always non-negative. Thus, the sequence $(x_k)_{k\in\mathbb{N}}$ has to be bounded. Since ∇f is locally smooth, it is smooth on bounded sets. This means that for the set

$\mathcal{C} = \overline{\text{conv}}\{x^*, x_0, x_1, \ldots\}$, which is bounded as the convex hull of bounded points, there exists $L_C > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L_C \|x - y\| \quad \forall x, y \in \mathcal{C}.$$

Clearly, $\lambda_1 = \frac{\|x_1 - x_0\|}{2\|\nabla f(x_1) - \nabla f(x_0)\|} \geqslant \frac{1}{2L}$, thus, by induction, it follows that $\lambda_k \geqslant \frac{1}{2L}$. This means the sequence $(\lambda_k)_{k\in\mathbb{N}}$ will never reach 0.
By L-smoothness (1.14),we get

$$\lambda_k \left(f(x^*) - f(x_k)\right) \geqslant \lambda_k \langle \nabla f(x_k), x^* - x_k \rangle + \frac{\lambda_k}{2L} \|\nabla f(x_k)\|^2. \tag{2.21}$$

Using this inequality instead of (2.5) in Lemma 2.1.2 yields for all $k \geqslant 1$

$$\begin{aligned} \|x_{k+1} - x^*\|^2 + \frac{1}{2}\|x_{k+1} - x_k\|^2 + 2\lambda_k \left(1 + \theta_k\right)\left(f(x_k) - f_*\right) + \frac{\lambda_k}{L}\|\nabla f(x_k)\|^2 \\ \leqslant \|x_k - x^*\|^2 + \frac{1}{2}\|x_k - x_{k-1}\|^2 + 2\lambda_k \theta_k \left(f(x_{k-1}) - f_*\right). \end{aligned} \tag{2.22}$$

Telescoping (2.22) implies that $\sum_{i=1}^{k} \frac{\lambda_i}{L} \left\|\nabla f\left(x_i\right)\right\|^2 \leqslant D$, and since $\lambda_i \geqslant \frac{1}{2L}$, we have $\lim_{k\to\infty} \nabla f(x_k) = 0$. We now have to show that $(x_k)_{k\in\mathbb{N}}$ converges. Let $\mathcal{X}$ be the set of solutions for (1.1) and let $a_k := \frac{1}{2}\left\|x_k - x_{k-1}\right\|^2 + 2\lambda_k \theta_k \left(f\left(x_{k-1}\right) - f_*\right)$. Applying Lemma 2.1.3 with this definition too (2.3) yields the result since $\lambda_{k+1}\theta_{k+1} \leqslant \left(1 + \theta_k\right)\lambda_k$. This proves that the sequence converges to an element in $\mathcal{X}$.
We will now focus on the rate of convergence.
Using the fact that $f(x) - f^*$ is convex we can apply Jensen's inequality to the sum of $f\left(x_i\right) - f_*$ in the left-hand side of (2.20) and see that,

$$\frac{D}{2S_k} \geqslant \frac{\lambda_k \left(1 + \theta_k\right)\left(f(x_k) - f_*\right) + \sum_{i=1}^{k-1} w_i \left(f(x_i) - f_*\right)}{S_k} \geqslant f(\hat{x}^k) - f_*. \tag{2.23}$$

Since $\frac{1}{2L} \leqslant \lambda_k$ holds for all $k \geqslant 1$, we see that

$$S_k = \sum_{i=1}^{k} \lambda_i + \lambda_1 \theta_1 \geqslant \frac{k}{2L}, \tag{2.24}$$

holds and therefore

$$f(\hat{x}_k) - f_* \leqslant \frac{D}{2S_k} \leqslant \frac{DL}{k}. \tag{2.25}$$

□

This means that we established the same theoretical rate of convergence as (1.8.3).

Remark 2.1.6. *Note that in practice λ_k is usually much larger than $\frac{1}{2L}$, since it depends on the local Lipschitz smoothness. It can therefore yield much better convergence than the theoretical bound would suggest.*

We use strong convexity again to prove a stronger convergence result. In order to deduce the result, we will additionally have a stricter restriction for λ_k. Let us therefore define the slightly changed update rule for λ_k as

$$\lambda_k = \min\left\{\sqrt{1+\frac{\theta_{k-1}}{2}\lambda_{k-1}}, \frac{\|x_k - x_{k-1}\|}{2\|\nabla f(x_k) - \nabla f(x_{k-1})\|}\right\}. \tag{2.26}$$

Theorem 2.1.7. *Let $f:\mathbb{R}^n \to \mathbb{R}$ be locally μ-strongly convex and locally L-smooth, then the sequence $(x_k)_{k\in\mathbb{N}}$ generated by algorithm 2.1.1 with the change* (2.26)*, converges to the solution x^* of the minimization problem* (1.1)*. The complexity to get $\|x_k - x^*\|^2 \leqslant \varepsilon$ is $\mathcal{O}\left(\kappa \log \frac{1}{\varepsilon}\right)$, where $\kappa = \frac{L}{\mu}$.*

Proof. Using the stricter bound (2.26) does not change the statement of Theorem 2.1.5, and therefore we get convergence and boundedness of $\mathcal{C} = \overline{\text{conv}}\{x^*, x_0, x_1, \ldots\}$. This means that there exist L and μ such that $f:\mathbb{R}^n \to \mathbb{R}$ is μ-strongly convex and L-smooth on $\mathcal{C}$. Strong convexity on $\mathcal{C}$ implies $\|\nabla f(x_k) - \nabla f(x_{k-1})\| \geqslant \mu\|x_k - x_{k-1}\|$, and therefore $\lambda_k \leqslant \frac{1}{2\mu}$ for all $k \geqslant 1$. By the definition of μ-strong convexity, we get

$$\lambda_k\langle \nabla f(x_k), x^* - x_k\rangle \leqslant \lambda_k\left(f(x^*) - f(x_k)\right) - \lambda_k\frac{\mu}{2}\|x^* - x_k\|^2. \tag{2.27}$$

L-smoothness together with the fact that $\lambda_k \leqslant \frac{1}{2\mu}$ implies

$$\begin{aligned}\lambda_k\langle \nabla f(x_k), x^* - x_k\rangle &\leqslant \lambda_k\left(f(x^*) - f(x_k)\right) - \lambda_k\frac{1}{2L}\|\nabla f(x_k)\|^2 \\ &= \lambda_k\left(f(x^*) - f(x_k)\right) - \frac{1}{2L\lambda_k}\|x_{k+1} - x_k\|^2 \\ &\leqslant \lambda_k\left(f(x^*) - f(x_k)\right) - \frac{\mu}{L}\|x_{k+1} - x_k\|^2.\end{aligned} \tag{2.28}$$

These two inequalities yield

$$\begin{aligned}\lambda_k\langle \nabla f(x_k), x^* - x_k\rangle &= \frac{1}{2}\lambda_k\langle \nabla f(x_k), x^* - x_k\rangle + \frac{1}{2}\lambda_k\langle \nabla f(x_k), x^* - x_k\rangle \\ &\leqslant \lambda_k\left(f(x^*) - f(x_k)\right) - \lambda_k\frac{\mu}{4}\|x_k - x^*\|^2 - \frac{\mu}{2L}\|x_{k+1} - x_k\|^2.\end{aligned} \tag{2.29}$$

Using this inequality in (2.4) yields

$$\begin{aligned}\|x_{k+1} - x^*\|^2 \leqslant \|x_k - x^*\|^2 &- 2\lambda_k(f(x_k) - f(x^*)) \\ &- \lambda_k\frac{\mu}{2}\|x_k - x^*\|^2 - \frac{\mu}{L}\|x_{k+1} - x_k\|^2 + \|x_{k+1} - x_k\|^2.\end{aligned} \tag{2.30}$$

We can use (2.11) from Lemma 2.1.2 which stated

$$\|x_{k+1} - x_k\|^2 \leqslant \frac{1}{2}\|x_k - x_{k-1}\|^2 - \frac{1}{2}\|x_{k+1} - x_k\|^2 + 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)). \tag{2.31}$$

Using (2.31) in (2.30) and rearranging the terms, yields

$$\begin{aligned}&\|x_{k+1}-x^*\|^2+\frac{1}{2}\left(1+\frac{2\mu}{L}\right)\|x_{k+1}-x_k\|^2+2\lambda_k\left(1+\theta_k\right)\left(f(x_k)-f(x^*)\right)\\&\leqslant\left(1-\frac{\lambda_k\mu}{2}\right)\|x_k-x^*\|^2+\frac{1}{2}\|x_k-x_{k-1}\|^2+2\lambda_k\theta_k\left(f(x_{k-1})-f(x^*)\right).\end{aligned}\tag{2.32}$$

By (2.26) we have $\lambda_k\theta_k\leqslant\lambda_{k-1}\left(1+\frac{\theta_{k-1}}{2}\right)$, and therefore

$$\begin{aligned}&\|x_{k+1}-x^*\|^2+\frac{1}{2}\left(1+\frac{2\mu}{L}\right)\|x_{k+1}-x_k\|^2+2\lambda_k\left(1+\theta_k\right)\left(f(x_k)-f_*\right)\\&\leqslant\left(1-\frac{\lambda_k\mu}{2}\right)\|x_k-x^*\|^2+\frac{1}{2}\|x_k-x_{k-1}\|^2+2\lambda_{k-1}\left(1+\frac{\theta_{k-1}}{2}\right)\left(f(x_{k-1})-f_*\right).\end{aligned}\tag{2.33}$$

Defining $\psi^{k+1}:=\|x_{k+1}-x^*\|^2+\frac{1}{2}\left(1+\frac{2\mu}{L}\right)\|x_{k+1}-x_k\|^2+2\lambda_k\left(1+\theta_k\right)\left(f(x_k)-f_*\right)$, and $\alpha_k:=1-\min\left\{\frac{\lambda_k\mu}{2},\frac{2\mu}{L+2\mu},\frac{\theta_{k-1}}{2(1+\theta_{k-1})}\right\}=\max\left\{1-\frac{\lambda_k\mu}{2},\frac{L}{L+2\mu},\frac{1+\frac{\theta_{k-1}}{2}}{1+\theta_{k-1}}\right\}$.
By (2.32), we therefore have

$$\begin{aligned}\psi^{k+1}&\leqslant\left(1-\frac{\lambda_k\mu}{2}\right)\|x_k-x^*\|^2+\frac{1}{2}\left(\frac{L}{L+2\mu}\right)\left(1+\frac{2\mu}{L}\right)\|x_k-x_{k-1}\|^2\\&\quad+2\lambda_{k-1}\left(\frac{1+\frac{\theta_{k-1}}{2}}{1+\theta_{k-1}}\right)\left(\theta_{k-1}+1\right)\left(f(x_{k-1})-f_*\right)\\&\leqslant\alpha_k\left(\|x_k-x^*\|^2+\frac{1}{2}\|x_k-x_{k-1}\|^2+2\lambda_{k-1}(1+\theta_{k-1})\left(f(x_{k-1})-f_*\right)\right)=\alpha_k\psi^k.\end{aligned}\tag{2.34}$$

We will now bound α_k. By the definition of κ, we have $\frac{2\mu}{L+2\mu}=\frac{2}{\kappa+2}\geqslant\frac{1}{4\kappa}$. Furthermore we know that $\frac{1}{2L}\leqslant\lambda_k$ for $k\geqslant1$, and therefore $\frac{\lambda_k\mu}{2}\geqslant\frac{1}{4\kappa}$ for $k\geqslant1$. Using the fact that $\frac{1}{2L}\leqslant\lambda_k\leqslant\frac{1}{2\mu}$ for $k>1$, we see that $\theta_k=\frac{\lambda_k}{\lambda_{k-1}}\geqslant\frac{1}{\kappa}$ for $k>1$. Therefore $\frac{\theta_{k-1}}{2(1+\theta_{k-1})}\geqslant\frac{1}{2(\kappa+1)}\geqslant\frac{1}{4\kappa}$, since $g(\theta):=\frac{\theta}{1+\theta}$ is monotonically increasing for $\theta>0$. We get that $\alpha_k\leqslant\left(1-\frac{1}{4\kappa}\right)$, and therefore $\psi^{k+1}\leqslant\left(1-\frac{1}{4\kappa}\right)\psi^k$. We now have $\|x_{k+1}-x^*\|^2\leqslant\left(1-\frac{1}{4\kappa}\right)^k C_0$. Therefore we get $\mathcal{O}\left(\kappa\log\frac{1}{\varepsilon}\right)$ complexity in order to achieve ϵ accuracy. □

Remark 2.1.8. *This complexity bound coincides with the one for the constant stepsize approach.*

2.2 A more general update rule

We will now see that the two bounds (2.1) and (2.2) can be generalized.

Algorithm 2.2.1: General Adaptive Gradient Descent

Input: A starting point $x_0 \in \mathbb{R}^n$, an initial stepsize $\lambda_0 > 0, \theta_0 = +\infty, \alpha \in (0,1)$ and $\beta = \frac{1}{2(1-\alpha)}$.

Routine:

$k := 1$

$x_1 = x_0 - \lambda_0 \nabla f(x_0)$

while *stopping criterion is not satisfied* **do**

  $\lambda_k = \min\left\{\sqrt{\frac{1}{\beta} + \theta_{k-1}}\lambda_{k-1}, \frac{\alpha\|x_k - x_{k-1}\|}{\|\nabla f(x_k) - \nabla f(x_{k-1})\|}\right\}$

  $x_{k+1} = x_k - \lambda_k \nabla f(x_k)$

  $\theta_k = \frac{\lambda_k}{\lambda_{k-1}}$

  $k = k + 1$

We will deduce a generalization of Lemma 2.1.2

Lemma 2.2.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and differentiable. Denote x^* as any solution of the minimization problem* (1.1)*. Then the following inequality holds for any x_k generated by algorithm 2.2.1*

$$\begin{aligned}&\|x_{k+1} - x^*\|^2 + 2\lambda_k(1 + \theta_k\beta)(f(x_k) - f_*) + \alpha\beta\|x_{k+1} - x_k\|^2 \\ &\leqslant \|x_k - x^*\|^2 + 2\lambda_k\theta_k\beta(f(x_{k-1}) - f_*) + \alpha\beta\|x_k - x_{k-1}\|^2 \quad \forall k \geqslant 1.\end{aligned} \tag{2.35}$$

Proof. Note that (2.7),(2.8) and (2.10) hold for any choice of λ_k. We state them again for convenience below.

$$\|x_{k+1} - x^*\|^2 \leqslant \|x_k - x^*\|^2 - 2\lambda_k(f(x_k) - f_*) + \|x_{k+1} - x_k\|^2 \quad \forall k \geqslant 1. \tag{2.36}$$

$$\begin{aligned}\|x_{k+1} - x_k\|^2 = {}& 2\lambda_k\langle\nabla f(x_k) - \nabla f(x_{k-1}), x_k - x_{k+1}\rangle \\ &+ 2\lambda_k\langle\nabla f(x_{k-1}), x_k - x_{k+1}\rangle - \|x_{k+1} - x_k\|^2 \quad \forall k \geqslant 1.\end{aligned} \tag{2.37}$$

$$2\lambda_k\langle\nabla f(x_{k-1}), x_k - x_{k+1}\rangle \leqslant 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)) \quad \forall k \geqslant 1. \tag{2.38}$$

Let $k \geqslant 1$. Combining (2.37) with (2.38) and applying the Cauchy–Schwarz inequality, we get

$$\begin{aligned}\|x_{k+1} - x_k\|^2 \leqslant {}& 2\lambda_k\|\nabla f(x_k) - \nabla f(x_{k-1})\|\|x_k - x_{k+1}\| \\ &+ 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)) - \|x_{k+1} - x_k\|^2\end{aligned} \tag{2.39}$$

Using the the new definition for λ_k, we get $\lambda_k\|\nabla f(x_k) - \nabla f(x_{k-1})\| \leqslant \alpha\|x_k - x_{k-1}\|$, and therefore

$$\|x_{k+1} - x_k\|^2 \leqslant 2\alpha\|x_k - x_{k-1}\|\|x_{k+1} - x_k\| + 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)) - \|x_{k+1} - x_k\|^2. \tag{2.40}$$

Applying Young's inequality to the first term, we get

$$\|x_{k+1} - x_k\|^2 \leqslant \alpha\|x_k - x_{k-1}\|^2 + \alpha\|x_{k+1} - x_k\|^2 + 2\lambda_k\theta_k(f(x_{k-1}) - f(x_k)) - \|x_{k+1} - x_k\|^2, \tag{2.41}$$

which, after multiplying both sides by β and rearranging the terms, yields

$$\beta(2-\alpha)\|x_{k+1}-x_k\|^2 + 2\beta\lambda_k\theta_k\left(f(x_k)-f_*\right) \leqslant \alpha\beta\|x_k-x_{k-1}\|^2 + 2\beta\lambda_k\theta_k\left(f(x_{k-1})-f_*\right). \tag{2.42}$$

Adding (2.36) to this equation, gives us

$$\begin{aligned}\|x_{k+1}-x^*\|^2 + 2\lambda_k\left(1+\theta_k\beta\right)\left(f(x_k)-f_*\right) + (2\beta-\alpha\beta-1)\|x_{k+1}-x_k\|^2 \\ \leqslant \|x_k-x^*\|^2 + 2\lambda_k\theta_k\beta\left(f(x_{k-1})-f_*\right) + \alpha\beta\|x_k-x_{k-1}\|^2.\end{aligned} \tag{2.43}$$

Since $\beta = \frac{1}{2(1-\alpha)}$, we have $2\beta-\alpha\beta-1 = \alpha\beta$, and therefore

$$\begin{aligned}\|x_{k+1}-x^*\|^2 + 2\lambda_k\left(1+\theta_k\beta\right)\left(f(x_k)-f_*\right) + \alpha\beta\|x_{k+1}-x_k\|^2 \\ \leqslant \|x_k-x^*\|^2 + 2\lambda_k\theta_k\beta\left(f(x_{k-1})-f_*\right) + \alpha\beta\|x_k-x_{k-1}\|^2.\end{aligned} \tag{2.44}$$

□

In order to prove convergence of algorithm 2.2.1, we need the following lemma.

Lemma 2.2.3. *If $f:\mathbb{R}^n\to\mathbb{R}$ is convex and locally L-smooth, then the sequence $(\lambda_k)_{k\in\mathbb{N}}$ generated by algorithm 2.2.1 satisfies the following inequality*

$$\text{if } \alpha \leqslant \frac{1}{2}, \text{ we have } \lambda_k \geqslant \frac{\alpha}{L} \text{ for all } k \geqslant 1, \tag{2.45}$$

$$\text{if } \alpha > \frac{1}{2}, \text{ we have } \lambda_k \geqslant \frac{2\alpha(1-\alpha)}{L} \text{ for all } k \geqslant 1. \tag{2.46}$$

Proof. Let $k\geqslant 1$, by the definition of the stepsize, we get $\lambda_i\left(1+\theta_i\beta\right)-\lambda_{i+1}\theta_{i+1}\beta \geqslant 0$. By (2.35) we see that $(x_k)_{k\in\mathbb{N}}$ is bounded, and since ∇f is locally Lipschitz, it is Lipschitz continuous on bounded sets. Let L denote the L-smoothness constant on the bounded set $\mathcal{C} = \overline{\operatorname{conv}}\{x^*, x_1, x_2, \ldots\}$. Note that $\lambda_1 = \frac{\alpha\|x_1-x_0\|}{\|\nabla f(x_1)-\nabla f(x_0)\|} \geqslant \frac{\alpha}{L}$ holds for both cases. In the first case, we have $\frac{1}{\beta} > 1$ and therefore, by induction, it follows that $\lambda_k \geqslant \frac{\alpha}{L}$ for all k. For the second case let m and n be the smallest numbers such that $\beta^{-\frac{1}{2^m}} \geqslant 1-\frac{1}{2\beta}$ and $(1+\frac{1}{\beta})^{\frac{n}{2}} \geqslant \beta$. If $\lambda_k \geqslant \frac{\alpha}{L}$ for all k, we are done. Therefore, in the other case, there exists a k such that $\lambda_{k-1} \geqslant \frac{\alpha}{L}$ and $\lambda_k < \frac{\alpha}{L}$. Now choose the largest j (possibly infinity) such that the second bound is not active for λ_{k+i} with $0\leqslant i<j$ and therefore $\lambda_{k+i} = \sqrt{\frac{1}{\beta}+\theta_{k+i-i}}\lambda_{k+i-1}$. This yields $\theta_{k+i} = \sqrt{\frac{1}{\beta}+\theta_{k+i-1}}$ for $0\leqslant i<j$ and we have

$$\theta_k \geqslant \beta^{-\frac{1}{2}}, \quad \theta_{k+1} \geqslant \sqrt{\frac{1}{\beta}+\sqrt{\frac{1}{\beta}}} \geqslant \beta^{-\frac{1}{4}}, \quad \ldots, \quad \theta_{k+i} \geqslant \sqrt{\frac{1}{\beta}+\beta^{-\frac{1}{2^i}}} \geqslant \beta^{-\frac{1}{2^{i+1}}} \tag{2.47}$$

for $0\leqslant i<j$. This yields

$$\frac{\lambda_{k+i}}{\lambda_{k-1}} = \theta_k\theta_{k+1}\ldots\theta_{k+i} \geqslant \beta^{-\frac{1}{2}-\frac{1}{4}-\cdots-\frac{1}{2^{i+1}}} \geqslant \beta^{-1} = 2(1-\alpha) \tag{2.48}$$

and therefore $\lambda_{k+i} \geqslant 2(1-\alpha)\lambda_{k-1} \geqslant \frac{2\alpha(1-\alpha)}{L}$ for $0\leqslant i<j$. If j is not infinite we can start the same argument again, since in this case at the $k+j$ iterate the second bound is active and we have $\lambda_{k+j} \geqslant \frac{\alpha}{L}$. □

Convergence

The proof that follows employs the same strategy as Theorem 2.1.7

Theorem 2.2.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and locally L-smooth. Let f_* denote the minimal value and x^* denote any solution of the minimization problem* (1.1) *assuming that the problem has a solution. Then the sequence $(x_k)_{k\in\mathbb{N}}$ generated by algorithm 2.2.1 converges to a solution of the minimization problem* (1.1)*, and we have that*

$$f(\hat{x}_k) - f_* \leqslant \frac{D}{2S_k} = \mathcal{O}\left(\frac{1}{k}\right) \quad \forall k \geqslant 1, \tag{2.49}$$

where

$$\begin{aligned}
\hat{x}_k &= \frac{\lambda_k\left(1+\theta_k\beta\right)x_k + \sum_{i=1}^{k-1} w_i x_i}{S_k},\\
w_i &= \lambda_i\left(1+\theta_i\beta\right) - \lambda_{i+1}\theta_{i+1}\beta,\\
S_k &= \lambda_k\left(1+\theta_k\beta\right) + \sum_{i=1}^{k-1} w_i = \sum_{i=1}^{k}\lambda_i + \lambda_1\theta_1\beta.
\end{aligned}$$

Proof. Let $k \geqslant 1$. Telescoping (2.35) yields

$$\begin{aligned}
&\|x_{k+1} - x^*\|^2 + \alpha\beta\|x_{k+1} - x_k\|^2 + 2\lambda_k(1+\theta_k\beta)(f(x_k) - f_*)\\
&\qquad +2\sum_{i=1}^{k-1}\left[\lambda_i\left(1+\theta_i\beta\right) - \lambda_{i+1}\theta_{i+1}\beta\right]\left(f\left(x_i\right) - f_*\right)\\
&\leqslant \left\|x_1 - x^*\right\|^2 + \alpha\beta\left\|x_1 - x_0\right\|^2 + 2\lambda_1\theta_1\beta\left[f\left(x_0\right) - f_*\right] =: D
\end{aligned} \tag{2.50}$$

Using the fact that $f(x) - f^*$ is convex, we can apply Jensen's inequality to the sum of $f(x_i) - f^*$ in the left-hand side of (2.50) and see that

$$\frac{D}{2S_k} \geqslant \frac{\lambda_k(1+\theta_k\beta)(f(x_k) - f_*) + \sum_{i=1}^{k-1} w_i\left(f(x_i) - f_*\right)}{S_k} \geqslant f(\hat{x}_k) - f_*. \tag{2.51}$$

For $\alpha \leqslant \frac{1}{2}$ by (2.2.3), we have $S_k = \sum_{i=1}^{k}\lambda_i + \lambda_1\theta_1\beta \geqslant \frac{k\alpha}{L}$, which implies

$$f(\hat{x}_k) - f_* \leqslant \frac{DL}{2k\alpha}. \tag{2.52}$$

For $\alpha > \frac{1}{2}$ by (2.2.3), we have $S_k = \sum_{i=1}^{k}\lambda_i + \lambda_1\theta_1\beta \geqslant \frac{k2\alpha(1-\alpha)}{L}$, which implies

$$f(\hat{x}_k) - f_* \leqslant \frac{DL}{4k\alpha(1-\alpha)}. \tag{2.53}$$

The proof for convergence now follows the same argument as in Theorem 2.3. Using L-smoothness we get

$$\lambda_k\left(f(x^*) - f(x_k)\right) \geqslant \lambda_k\langle\nabla f(x_k), x^* - x_k\rangle + \frac{\lambda_k}{2L}\|\nabla f(x_k)\|^2, \tag{2.54}$$

which can be used in Lemma 2.8 and yields

$$\begin{aligned}&\|x_{k+1}-x^*\|^2+2\lambda_k\left(1+\theta_k\beta\right)\left(f(x_k)-f_*\right)+\alpha\beta\|x_{k+1}-x_k\|^2+\frac{\lambda_k}{L}\|\nabla f(x_k)\|^2\\&\leqslant\|x_k-x^*\|^2+2\lambda_k\theta_k\beta\left(f(x_{k-1})-f_*\right)+\alpha\beta\|x_k-x_{k-1}\|^2.\end{aligned}\tag{2.55}$$

Telescoping this inequality implies that $\sum_{i=1}^{k}\frac{\lambda_i}{L}\left\|\nabla f\left(x_i\right)\right\|^2\leqslant D$ for all k, and therefore $\lim_{k\to\infty}\nabla f\left(x_k\right)=0$. Since $\{\lambda_k\}_{k\in\mathbb{N}}$ is not a null sequence by (2.2.3). Applying Lemma 2.2 to Lemma 2.8 and using the fact that $\lambda_i\left(1+\theta_i\beta\right)\geqslant\lambda_{i+1}\theta_{i+1}\beta$, implies that $\lim_{k\to\infty}x_k$ exists. □

2.3 Adaptive gradient descent with known smoothness constant

If the L-smoothness constant is known, another update rule was proposed in the original paper [9].

Algorithm 2.3.1: Adaptive Gradient with known L

Input: A starting point $x_0\in\mathbb{R}^n$, an initial stepsize $\lambda_0>0, x_0\in\mathbb{R}^d, \lambda_0=\frac{1}{L}, \theta_0=+\infty$.

Routine:

$k=1$
$x_1=x_0-\lambda_0\nabla f\left(x_0\right)$
while *stopping criterion is not satisfied* **do**
  $L_k=\frac{\left\|\nabla f\left(x^k\right)-\nabla f\left(x^{k-1}\right)\right\|}{\|x^k-x^{k-1}\|}$
  $\lambda_k=\min\left\{\sqrt{1+\theta_{k-1}}\lambda_{k-1},\frac{1}{\lambda_{k-1}L^2}+\frac{1}{2L_k}\right\}$
  $x_{k+1}=x_k-\lambda_k\nabla f(x_k)$
  $\theta_k=\frac{\lambda_k}{\lambda_{k-1}}$
  $k=k+1$

Theorem 5 in [9] contains the proof for the following theorem.

Theorem 2.3.1. *Let $f:\mathbb{R}^n\to\mathbb{R}$ be convex and locally L-smooth. Then Lemma 2.1.2 holds for $(x_k)_{k\in\mathbb{N}}$ generated by* (2.3.1) *as well.*

3 Stochastic gradient descent methods

3.1 Stochastic optimization

We are now trying to solve the following stochastic optimization problem

$$\min_{x\in\mathbb{R}^n}\{F(x) := \mathbb{E}[f(x;\xi)] := \mathbb{E}[f_\xi(x)]\}, \tag{3.1}$$

where ξ is a random variable following some unknown distribution, $f_\xi(x)$ denotes a differentiable function $f_\xi : \mathbb{R}^n \to \mathbb{R}$ and $\mathbb{E}$ denotes the expected value in regards of the unknown distribution. In a similar fashion, when given a set of realizations ξ^i of ξ, we can define the empirical risk minimization problem as

$$\min_{x\in\mathbb{R}^n}\left\{F(x) = \frac{1}{n}\sum_{i=1}^{n} f_{\xi^i}(x)\right\}. \tag{3.2}$$

For further details see [13].

Stochastic gradient descent

Stochastic gradient descent is a well-known approach to this problem, but once again, finding the right step size is necessary.

Algorithm 3.1.1: SGD algorithm constant stepsize

Input: $x_0 \in \mathbb{R}^n$ and $\lambda > 0$.

Routine:

1 $k := 0$
2 **while** *stopping criterion is not satisfied* **do**
3 Sample ξ^k
4 Compute a stochastic vector $g(x_k;\xi^k)$
5 $x_{k+1} := x_k - \lambda g(x_k;\xi^k)$
6 $k = k + 1$

Possible choices for the stochastic vector are

$$g\left(x_k,\xi_k\right) = \begin{cases} \nabla f\left(x_k;\xi^k\right) \\ \frac{1}{n_k}\sum_{i=1}^{n_k}\nabla f\left(x_k;\xi^i\right) \end{cases}, \tag{3.3}$$

where the first choice is known as simple SG and the second choice is known as mini-batch SG. Theorem 4.6 in [4] provides a necessary condition for algorithm 3.1.1 convergence in expectation.

Adaptive stochastic gradient descent unbiased

In their paper [9], Yura Malitsky and Konstantin Mishchenko proposed the following version.

Algorithm 3.1.2: Adaptive Stochastic Gradient Descent unbiased

Input: $x_0 \in \mathbb{R}^d, \lambda_0 > 0, \theta_0 = +\infty, \xi^0$and $\alpha > 0$.

Routine:

$k := 0$

while *stopping criterion is not satisfied* **do**

- Sample ξ^k and ζ^k
- $L_k := \frac{\|\nabla f_{\zeta^k}(x_k) - \nabla f_{\zeta^k}(x_{k-1})\|}{\|x_k - x_{k-1}\|}$
- $\lambda_k := \min\left\{\sqrt{1+\theta_{k-1}}\lambda_{k-1}, \frac{\alpha}{L_k}\right\}$
- $x_{k+1} := x_k - \lambda_k \nabla f_{\xi^k}(x_k)$
- $\theta_k := \frac{\lambda_k}{\lambda_{k-1}}$
- $k = k + 1$

We will now start our analysis for the proposed algorithm.

Remark 3.1.3. *If $f_\xi : \mathbb{R}^n \to \mathbb{R}$ is almost surely L-smooth, $F : \mathbb{R}^n \to \mathbb{R}$ defined as in* (3.1) *is L-smooth with the same constant L.*

Lemma 3.1.4. *Let $f_\xi : \mathbb{R}^n \to \mathbb{R}$ be L-smooth and μ-strongly convex almost surely. It holds for λ_k produced by algorithm 3.1.2*

$$\frac{\alpha}{L} \leqslant \lambda_k \leqslant \frac{\alpha}{\mu} \quad \text{a.s.} \tag{3.4}$$

Proof. Strong convexity together with the Cauchy–Schwarz inequality implies $\|x - y\| \leqslant \frac{1}{\mu}\|\nabla f_{\zeta^k}(x) - \nabla f_{\zeta^k}(y)\|$ for any x, y. Therefore, $\lambda_k \leqslant \min\left\{\sqrt{1+\theta_k}\lambda_{k-1}, \frac{\alpha}{\mu}\right\} \leqslant \frac{\alpha}{\mu}$ a.s. L-smoothness gives, by definition $\frac{\|\nabla f_{\zeta^k}(x) - \nabla f_{\zeta^k}(y)\|}{\|x-y\|} \leqslant L$ for any x, y, which means that $\lambda_k \geqslant \min\left\{\sqrt{1+\theta_k}\lambda_{k-1}, \frac{\alpha}{L}\right\} \geqslant \min\{\lambda_{k-1}, \frac{\alpha}{L}\}$ a.s. The lower bound is now achieved by iterating this inequality, since we get $\lambda_k \geqslant \min\{\lambda_1, \frac{\alpha}{L}\}$ a.s. and $\lambda_1 := \frac{\alpha}{L1} \geqslant \frac{\alpha}{L}$. □

Lemma 3.1.5. *Let F_* and x^* denote the optimal value and the minimum of* (3.1). *Furthermore let $f_\xi : \mathbb{R}^n \to \mathbb{R}$ to be almost surely L-smooth and convex and define $\sigma^2 := \mathbb{E}\left[\|\nabla f_\xi(x^*)\|^2\right]$. Then the following inequality holds for any x*

$$\mathbb{E}\left[\|\nabla f_\xi(x)\|^2\right] \leqslant 4L\left(F(x) - F_*\right) + 2\sigma^2 \quad \text{a.s.} \tag{3.5}$$

Proof. We start with the following observation

$$\|a\|^2 = \|a - b + b\|^2 \leqslant 2\|a - b\|^2 + 2\|b\|^2, \tag{3.6}$$

which holds for any $a, b \in \mathbb{R}^n$.

Applying now (1.14) yields

$$\|\nabla f_\xi(x) - \nabla f_\xi(x^*)\|_2^2 \leqslant 2L\left[f_\xi(x) - f_\xi(x^*) - \langle \nabla f_\xi(x^*), x - x^*\rangle\right]. \tag{3.7}$$

Taking the expected value on both sides, and using the fact that $\nabla F(x^*) = 0$, we obtain

$$\mathbb{E}\left[\|\nabla f_\xi(x) - \nabla f_\xi(x^*)\|^2\right] \leqslant 2L\left[F(x) - F(x^*)\right]. \tag{3.8}$$

Therefore by (3.6) and (3.8),

$$\mathbb{E}\left[\|\nabla f_\xi(x)\|^2\right] \leqslant 4L\left[F(x) - F(x^*)\right] + 2\mathbb{E}\left[\|\nabla f_\xi(x^*)\|^2\right].$$

□

Theorem 3.1.6. *Let $f_\xi : \mathbb{R}^n \to \mathbb{R}$ be L-smooth and μ-strongly convex almost surely, and let x^* denote the minimum of (3.1). If we choose $\alpha \leqslant \frac{\mu}{2L}$, then the sequence $(x_k)_{k\in\mathbb{N}}$ generated by algorithm 3.1.2 satisfies*

$$\mathbb{E}\left[\|x_k - x^*\|^2\right] \leqslant e^{\left(-k\frac{\mu\alpha}{L}\right)}C_0 + 2\alpha\frac{\sigma^2}{\mu^2} \quad \forall k \geqslant 1$$

with constants $C := 2\left(1 + 2\lambda_0^2 L^2\right)\|x_0 - x^*\|^2 + 4\lambda_0^2\sigma^2$ and $\sigma^2 := \mathbb{E}\left[\|\nabla f_\xi(x^*)\|^2\right]$.

Proof. Let $k \geqslant 1$. Since $\alpha \leqslant \frac{\mu}{2L}$, we have $\lambda_k \leqslant \frac{\alpha}{\mu} \leqslant \frac{1}{2L}$. Furthermore since λ_k is independent of ξ^k, we have $\mathbb{E}\left[\lambda_k \nabla f_{\xi^k}(x_k)\right] = \mathbb{E}\left[\lambda_k\right]\mathbb{E}\left[\nabla f_{\xi^k}(x_k)\right]$. Therefore

$$\begin{aligned}
\mathbb{E}\Big[\|x_{k+1} - x^*\|^2\Big] &= \mathbb{E}\left[\|x_k - x^*\|^2\right] - 2\mathbb{E}\left[\lambda_k\langle \nabla f(x_k), x_k - x^*\rangle\right] + \mathbb{E}\left[\lambda_k^2\right]\mathbb{E}\left[\|\nabla f_{\xi^k}(x_k)\|^2\right] \\
&\overset{(1.5.2)}{\leqslant} \mathbb{E}\left[(1 - \lambda_k\mu)\|x_k - x^*\|^2\right] - 2\mathbb{E}\left[\lambda_k(f(x_k) - f(x^*))\right] + \mathbb{E}[\lambda_k^2]\mathbb{E}\left[\|\nabla f_{\xi^k}(x_k)\|^2\right] \\
&\overset{(3.5)}{\leqslant} \mathbb{E}\left[(1 - \lambda_k\mu)\|x_k - x^*\|^2\right] - 2\mathbb{E}[\lambda_k \underbrace{(1 - 2\lambda_k L)}_{\geqslant 0}(f(x_k) - f(x^*))] + 2\mathbb{E}\left[\lambda_k^2\right]\sigma^2 \\
&\overset{(3.4)}{\leqslant} \mathbb{E}\left[1 - \lambda_k\mu\right]\mathbb{E}\left[\|x_k - x^*\|^2\right] + 2\alpha\frac{\mathbb{E}\left[\lambda_k\right]\sigma^2}{\mu}.
\end{aligned} \tag{3.9}$$

Therefore, if we subtract $2\alpha\frac{\sigma^2}{\mu^2}$ from both sides, we obtain

$$\mathbb{E}\left[\|x_{k+1} - x^*\|^2 - 2\alpha\frac{\sigma^2}{\mu^2}\right] \leqslant \mathbb{E}\left[1 - \lambda_k\mu\right]\mathbb{E}\left[\|x_k - x^*\|^2 - 2\alpha\frac{\sigma^2}{\mu^2}\right]. \tag{3.10}$$

By (1.5.2), and (1.6.2) it follows that $\mu \leqslant L$, therefore $\lambda_k\mu \leqslant \frac{\mu}{2L} \leqslant \frac{1}{2}$. Using this fact and (3.10) we see that, if $\mathbb{E}[\|x^k - x^*\|^2] \leqslant 2\alpha\frac{\sigma^2}{\mu^2}$ for some k, it implies $\mathbb{E}[\|x^j - x^*\|^2] \leqslant 2\alpha\frac{\sigma^2}{\mu^2} \quad \forall j > k$. Otherwise we iterate inequality (3.10) and obtain

$$\mathbb{E}\left[\|x_{k+1} - x^*\|^2\right] \leqslant \prod_{t=1}^{k}\mathbb{E}\left[1 - \lambda_t\mu\right]\|x_1 - x^*\|^2 + 2\alpha\frac{\sigma^2}{\mu^2}. \tag{3.11}$$

Since $1 - x \leqslant e^{-x}$, we have

$$\prod_{t=1}^{k} \mathbb{E}\left[1 - \lambda_t \mu\right] = \mathbb{E}\left(\prod_{t=1}^{k} \left[1 - \lambda_t \mu\right]\right) \leqslant \mathbb{E}\left(e^{\left(-\mu \sum_{t=1}^{k} \lambda_t\right)}\right). \tag{3.12}$$

By (3.4) we have $\lambda_k \geqslant \frac{\alpha}{L}$. Thus,

$$\mathbb{E}\left(e^{\left(-\mu \sum_{t=0}^{k} \lambda_t\right)}\right) \leqslant e^{-k\alpha \frac{\mu}{L}}. \tag{3.13}$$

From this we get,

$$\mathbb{E}\left[\|x_{k+1} - x^*\|^2\right] \leqslant e^{\left(-k\alpha \frac{\mu}{L}\right)} \mathbb{E}\left[\|x_0 - x^*\|^2\right] + 2\alpha \frac{\sigma^2}{\mu^2}. \tag{3.14}$$

By (3.5) and L-smoothness of F we have,

$$\begin{aligned} \mathbb{E}\left[\|x_1 - x^*\|^2\right] &\leqslant 2\|x_0 - x^*\|^2 + 2\lambda_0^2 \mathbb{E}\left[\|\nabla f_{\xi^0}(x_0)\|^2\right] \\ &\leqslant 2\|x_0 - x^*\|^2 + 2\lambda_0^2 \left(4L(F(x_0) - F^*) + 2\sigma^2\right) \\ &\leqslant 2\|x_0 - x^*\|^2 + 2\lambda_0^2 \left(2L^2 \|x_0 - x^*\|^2 + 2\sigma^2\right). \end{aligned} \tag{3.15}$$

This finishes the proof. □

Adaptive stochastic gradient descent biased

Naturally, one might wonder what would happen if we did not take different samples for L_k and the gradient. This results in the algorithm shown below.

Algorithm 3.1.7: Adaptive Stochastic Gradient Descent Biased

Input: $x_0 \in \mathbb{R}^d, \lambda_0 > 0, \theta_0 = +\infty, \xi^0, \alpha > 0.$

Routine:

$k := 0$

$x_1 = x_0 - \lambda_0 \nabla f(x_0)$

while *stopping criterion is not satisfied* **do**

  Sample ξ^k

  $L_k := \frac{\|\nabla f_{\xi^k}(x_k) - \nabla f_{\xi^k}(x_{k-1})\|}{\|x_k - x_{k-1}\|}$

  $\lambda_k := \min\left\{\sqrt{1+\theta_{k-1}}\lambda_{k-1}, \frac{\alpha}{L_k}\right\}$

  $x_{k+1} := x_k - \lambda_k \nabla f_{\xi^k}(x_k)$

  $\theta_k := \frac{\lambda_k}{\lambda_{k-1}}$

  $k = k + 1$

Theorem 3.1.8. *Let $f_\xi : \mathbb{R}^n \to \mathbb{R}$ be L-smooth and μ-strongly convex almost surely, and let x^* denote the optimum of (3.1). Furthermore let the model be overparameterized, meaning that $\nabla f_\xi(x^*) = 0$ holds with probability one. If we choose $\alpha \leqslant \frac{\mu}{L}$, then the sequence $(x_k)_{k\in\mathbb{N}}$ generated by algorithm 3.1.7 satisfies*

$$\mathbb{E}\left[\|x_k - x^*\|^2\right] \leqslant e^{\left(-k\alpha\frac{\mu}{L}\right)} C_0 \quad \forall k \geqslant 1 \tag{3.16}$$

with $C_0 := 2\left(1 + \lambda_0^2 L^2\right)\|x_0 - x^*\|^2$.

Proof. Let $k \geqslant 1$. Since f_ξ is μ-strongly convex, we have

$$\langle \nabla f_{\xi^k}(x_k), x^* - x_k\rangle \leqslant f_{\xi^k}(x^*) - f_{\xi^k}(x_k) - \frac{\mu}{2}\|x_k - x^*\|^2 \quad \text{a.s.} \tag{3.17}$$

Using (1.14) and the fact that $\nabla f_\xi(x^*) = 0$ holds with probability 1, we see that

$$\mathbb{E}\left[\lambda_k^2 \|\nabla f_{\xi^k}(x_k)\|^2\right] \leqslant 2L\mathbb{E}\left[\lambda_k^2 (f_{\xi^k}(x_k) - f_{\xi^k}(x^*))\right], \tag{3.18}$$

.

Since $\alpha \leqslant \frac{\mu}{L}$ implies $\lambda_k \leqslant \frac{1}{L}$, we see that

$$\begin{aligned}
\mathbb{E}\|x_{k+1} - x^*\|^2 &= \mathbb{E}\|x_k - x^*\|^2 + 2\mathbb{E}[\lambda_k \langle \nabla f_{\xi^k}(x_k), x^* - x_k\rangle] + \mathbb{E}\left[\lambda_k^2 \|\nabla f_{\xi^k}(x_k)\|^2\right] \\
&\overset{(3.17)}{\leqslant} \mathbb{E}[(1 - \lambda_k\mu)]\mathbb{E}\left[\|x_k - x^*\|^2\right] - \mathbb{E}\left[2\lambda_k\left(f_{\xi^k}(x_k) - f_{\xi^k}(x^*)\right)\right] + \mathbb{E}\left[\lambda_k^2\|\nabla f_{\xi^k}(x_k)\|^2\right] \\
&\overset{(3.18)}{\leqslant} \mathbb{E}[(1 - \lambda_k\mu)]\mathbb{E}\left[\|x_k - x^*\|^2\right] - \mathbb{E}\left[2\lambda_k(1 - \lambda_k L)\left(f_{\xi^k}(x_k) - f_{\xi^k}(x^*)\right)\right] \\
&\overset{(\lambda_k \leqslant \frac{1}{L})}{\leqslant} \mathbb{E}[(1 - \lambda_k\mu)]\mathbb{E}\left[\|x_k - x^*\|^2\right]
\end{aligned} \tag{3.19}$$

We can now repeat the steps starting from (3.11) and get

$$\mathbb{E}\left[\|x_{k+1} - x^*\|^2\right] \leqslant e^{\left(-k\alpha\frac{\mu}{L}\right)}\mathbb{E}\left[\|x_1 - x^*\|^2\right]. \tag{3.20}$$

From L-smoothness we also get

$$\mathbb{E}[\|\nabla f_{\xi^0}(x_0)\|] = \mathbb{E}[\|\nabla f_{\xi^0}(x_0) - \nabla f_{\xi^0}(x^*)\|] \leqslant L\|x_0 - x^*\|. \tag{3.21}$$

Therefore

$$\mathbb{E}\left[\|x_1 - x^*\|^2\right] \leqslant 2\|x_0 - x^*\|^2 + 2\lambda_0^2\mathbb{E}\left[\|\nabla f_{\xi^0}(x_0)\|^2\right] \leqslant 2(1 + \lambda_0^2 L^2)\|x_0 - x^*\|. \tag{3.22}$$

□

A few numerical experiments will be examined in the following chapter.

4 Numerical experiments

4.1 Logistic Regression

We want to minimize the following function

$$\min_{x\in\mathbb{R}^n} f(x) = -\frac{1}{m}\sum_{i=1}^{m}\Big(y_i \log\left(s(a_i^\top x)\right) + (1-y_i)\log\left(1-s(a_i^\top x)\right)\Big) + \frac{\gamma}{2}\|x\|^2, \tag{4.1}$$

where $a_i \in \mathbb{R}^n, y_i \in \{0,1\}$,$s(z) = \frac{1}{1+\exp(-z)}$ is the sigmoid function. This is a convex and L-smooth problem.
Since $f_1(z) := -\log\left(s(z)\right)$ and $f_2(z) := -\log\left(1-s(z)\right)$ and their derivatives are given by $f_1'(z) := -1 + s(z)$ and $f_2'(z) := s(z)$. Therefore $f_1''(z) = f_2''(z) = \frac{e^z}{e^z+1} > 0$, which implies that they are convex. Furthermore, f as a sum of convex functions is convex. We calculate the gradient. The partial derivative is given as

$$\frac{\partial}{\partial x_j} f(x) = -\frac{1}{m}\sum_{i=1}^{m} y_i \frac{\partial}{\partial x_j}\left[\log(s(a_i^\top x)\right] + (1-y_i)\,\frac{\partial}{\partial x_j}\left[\log\left(1-s(a_i^\top x)\right)\right] + \frac{\gamma}{2}\frac{\partial}{\partial x_j}\|x\|^2 \tag{4.2}$$

$\forall j = 1...n$. Looking at each of the terms individually, we have

$$\begin{aligned}
\frac{\partial}{\partial x_j}\left[s(a_i^\top x)\right] &= s(a_i^\top x)\left(1-s(a_i^\top x)\right)[a_i]_j \quad \forall j = 1...n,\\
\frac{\partial}{\partial x_j}\left[\log(s(a_i^\top x)\right] &= \frac{1}{s(a_i^\top x)}\frac{\partial}{\partial x_j}\left[s(a_i^\top x)\right] \quad \forall j = 1...n,\\
\frac{\partial}{\partial x_j}\left[\log(1-s(a_i^\top x)\right] &= \frac{-1}{1-s(a_i^\top x)}\frac{\partial}{\partial x_j}\left[s(a_i^\top x)\right] \quad \forall j = 1...n.
\end{aligned} \tag{4.3}$$

Plugging (4.3) into (4.2), yields

$$\frac{\partial}{\partial x_j} f(x) = -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y_i}{s(a_i^\top x)} - \frac{1-y_i}{1-s(a_i^\top x)}\right] s(a_i^\top x)\left(1-s(a_i^\top x)\right)[a_i]_j + \gamma x_j \quad \forall j = 1...n, \tag{4.4}$$

which after simplifying the terms, yields

$$\frac{\partial}{\partial x_j} f(x) = \frac{1}{m}\sum_{i=1}^{m}\left(s(a_i^\top x) - y_i\right)[a_i]_j + \gamma x_j \quad \forall j = 1...n. \tag{4.5}$$

Therefore

$$\nabla f(x) = \frac{1}{m}\sum_{i=1}^{m} a_i \left(s(a_i^\top x) - y_i\right) + \gamma x. \tag{4.6}$$

The L-smoothness constant is derived from the mean value theorem, we know that there exists a $\xi \in (u, v)$ such that

$$|s(u) - s(v)| = |s'(\xi)||u - v|, \tag{4.7}$$

for all $u, v \in \mathbb{R}$. Since the derivative is bounded by $\frac{1}{4}$, we get

$$|s(u) - s(v)| \leqslant \frac{1}{4}|u - v| \tag{4.8}$$

for all $u, v \in \mathbb{R}$. Let $A := (a_1, ..., a_m)$

$$\begin{aligned}
\|\nabla f(x) - \nabla f(y)\| &\leqslant \frac{1}{m}\left\| A \begin{pmatrix} s(a_1^\top x) - y_1 \\ \vdots \\ s(a_m^\top x) - y_m \end{pmatrix} - A \begin{pmatrix} s(a_1^\top y) - y_1 \\ \vdots \\ s(a_m^\top y) - y_m \end{pmatrix} \right\| + \gamma\|x - y\| \\
&\leqslant \frac{1}{m}\|A\| \left\| \begin{pmatrix} s(a_1^\top x) - s(a_1^\top y) \\ \vdots \\ s(a_m^\top x) - s(a_m^\top y) \end{pmatrix} \right\| + \gamma\|x - y\| \\
&= \frac{1}{m}\|A\| \sqrt{\sum_{i=1}^{m} \left(s(a_i^\top x) - s(a_i^\top y)\right)^2} + \gamma\|x - y\| \\
&\overset{(4.8)}{\leqslant} \frac{1}{4m}\|A\| \sqrt{\sum_{i=1}^{m} \left(a_i^\top (x - y)\right)^2} + \gamma\|x - y\| \\
&= \frac{1}{4m}\|A\|\|A^T\|\|x - y\| + \gamma\|x - y\| = \left(\frac{1}{4m}\|A\|^2 + \gamma\right)\|x - y\|
\end{aligned} \tag{4.9}$$

Using the fact that $\|A\|^2 \leqslant \lambda_{max}(A^T A)$, we get that $L = \frac{1}{4m}\lambda_{max}(A^T A) + \gamma$. We use the Skin Segmentation Data Set from UCI [5] for our experiment. We compare the proposed algorithm 2.1.1 (ADGD) with constant stepsize gradient descent, Armijo-Line search, BB1, BB2, and Nesterov Accelerated Gradient Descent. For the constant stepsize, we use $\frac{1}{L}$ and get the following results. As seen in Figure 4.1.
We plot the ROC curve in Figure 4.2, which measures the true positive rate against the false positive rate. It is a measure of performance for a given binary classification model. We did another run with 1000 iterations and see that the model reaches its limit and all the algorithms yield the same accuracy in Table 4.1.

Figure 4.1: Error after 50 iterations

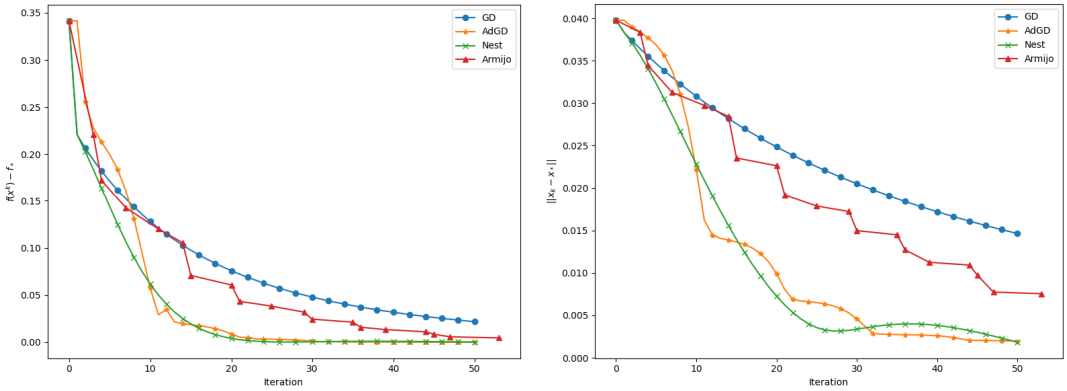
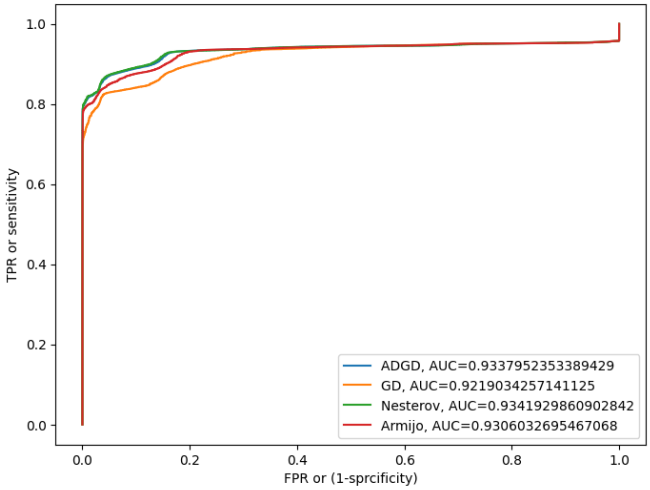


Table 4.1: Accuracy comparison

Algorithm	**Accuracy**
ADGD	0.9076
GD	0.9078
Nesterov	0.9076
Armijo	0.9076

Figure 4.2: ROC curve after 50 iterations



We will now generalize logistic regression to the multiclass case in order to observe its performance on a more complex problem.

4.2 Multiclass Regression

We want to minimize the following function

$$\min_{X\in\mathbb{R}^{n\times K}} L(X) = -\left[\sum_{i=1}^{m}\sum_{k=1}^{K} I\left\{y_i = k\right\} \log \frac{\exp\left(x_k^T a_i\right)}{\sum_{j=1}^{K}\exp\left(x_j^T a_i\right)}\right], \tag{4.10}$$

where $X = \begin{bmatrix} | & | & | & | \\ x_1 & x_2 & \cdots & x_K \\ | & | & | & | \end{bmatrix}, a_i \in \mathbb{R}^n, y_i \in \{1, ..., K\}$.

This loss function is known as cross-entropy for multiclass classification, and $\frac{\exp(z_k)}{\sum_{j=1}^{K}\exp(z_j)}$ is called the softmax function. The model uses a linear transformation before applying the softmax function, which can be interprated as probabilities for each class. The cross-entropy loss function measures now how far off we are from the true label. This loss function can be derived from the Kullback–Leibler divergence. We calculate the gradient.
Let $z_j = x_k^T a_j$ and $p_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$ it follows

$$\frac{\partial p_j}{\partial z_j} = \begin{cases} p_i\left(1 - p_j\right) & \text{if} \quad i = j \\ -p_j \cdot p_i & \text{if} \quad i \neq j \end{cases}, \tag{4.11}$$

from which we get

$$\frac{\partial L}{\partial z_i} = p_i - y_i. \tag{4.12}$$

Therefore

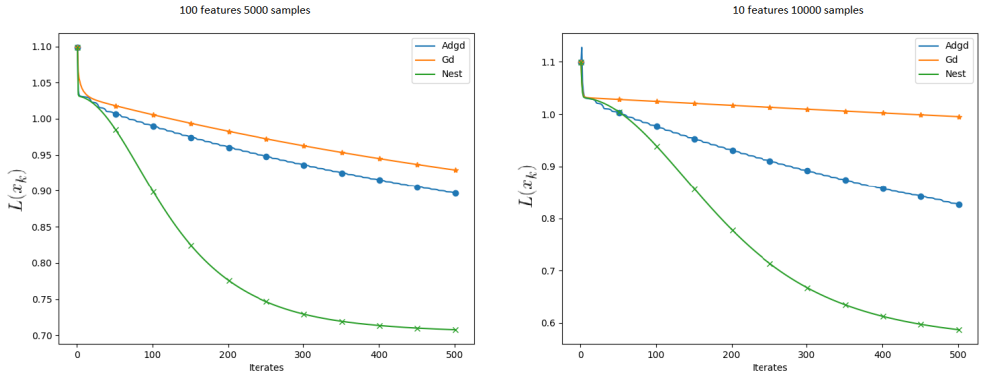
$$\frac{\partial L}{\partial x_i^{(j)}} = (p_i - y_i)a_i^{(j)}. \tag{4.13}$$

We don't use a real-world data set but rather generated the data ourselves with scikit-learn [14]. This allows for a variety of datasets to try the algorithm on.
We start with a dataset that has 100 features, 3 classes, and 5000 samples. Comparing regular gradient descent, Nesterov-accelerated gradient descent, and algorithm 2.1.1 (ADGD). We get the following results after averaging 10 randomly generated datasets.
The three algorithms did not significantly differ in terms of runtime. It should be noted, however, that the adaptive gradient descent produced descent results with no parameter adjustments. Furthermore, changes to the adaptive gradient descent algorithm's parameters had no effect on the outcome. Following another experiment with 10 features and 10000 samples. Nesterov's approach outperformed the others once more after some parameter adjustments. This can also be seen in the figure below.

Table 4.2: Accuracy Comparison

Algorithm	Accuracy average for 10 random datasets
GD	0.5606
ADGD	0.6016
Nesterov	0.7004

Figure 4.3: Loss Multiclass



4.3 Neural Network

In order to observe the behaviour of the proposed stochastic algorithm 3.1.7 we will look at an image classification problem. We will use the CIFAR-10 data-set [7], to cut down on computation time we use only half the data for the test-set and training-set. The training-set therefore consists of 25000 images, the test-set of 5000 and there are 10 different categories. We will use the "GoogLeNet" architecture to tackle this problem which won the ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) in the year 2014. The "GoogLeNet" structure is a deep convolutional neural network architecture for further information see [17]. We use once again the cross-entropy loss and therefore the minimization problem looks similar to the problems before but involves a more complicated model and therefore calculating the gradient is more involved, but it can be calculated in the same way as before which is known as back-propagation. We use Mini-batch gradient descent with a batch-size of 128 and let the algorithm run for 50 epochs. This means the algorithm split the training-set into subsets each containing 128 random samples (the last set contains less) 50 times and performed one step for each of these subsets, which explains the name Mini-batch. The implementation of the "GoogLeNet" was taken from a publicly available github repository [1]. The pytorch implementation for the biased version of the algorithm was taken from the github repository of the

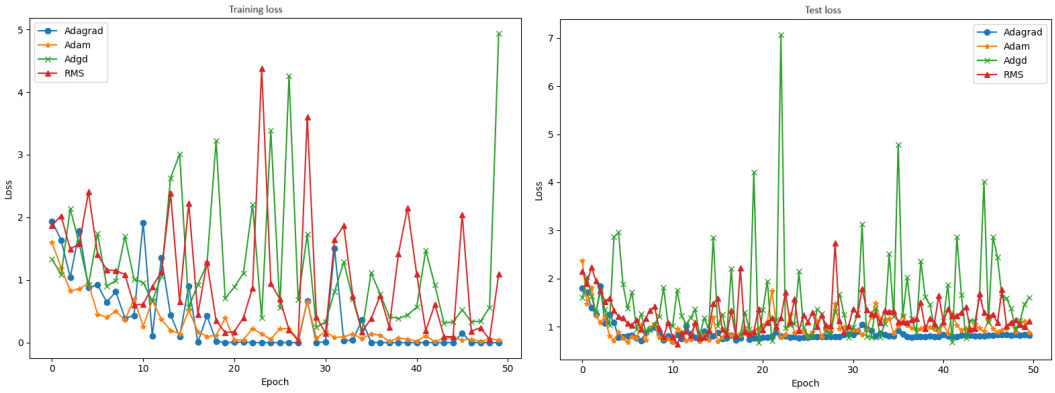
[1] `https://github.com/kuangliu/pytorch-cifar/blob/master/models/googlenet.py`

original paper[2]. For comparison we will use the three most commonly used algorithms for these kind of problems. Namely Adagrad, RMSprop and Adam. Adagrad, short for Adaptive Gradient Algorithm adapts the learning rate for each individual parameter of the model based on how often and how large the change to it was in the past. Since it updates the learning rate individual for each parameter it is a good candidate for sparse data-sets. Since Adagrad faced the problem that the learning-rates tend to decrease too fast RMSprop was developed. It tries to fix this issue by introducing a decay rate which controls how large the contribution of old gradients is. The most popular one is Adam and it can be seen as a combination of RMSprop and Adagrad. For a detailed explanation of these algorithms see [15].

[2] `https://github.com/ymalitsky/adaptive_GD/blob/master/pytorch/optimizer.py`

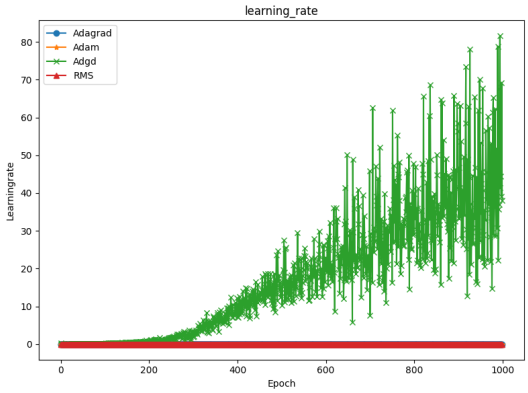
In order to make a fair comparison we fixed the randomness of the mini batch method by using a manual seed. We use the default parameters for Adam, Adagrad and RMSprop. For our algorithm we use $\alpha = 1$ and $\lambda_0 = 0.2$, and get the following results. We see that

Figure 4.4: Loss for training and test data



the algorithm performs rather bad in order to understand the cause of this we look at the stepsizes. We observe that the learning rates grow very fast, in order to fix this in hopes

Figure 4.5: Exploding gradient



of better performance we introduce a new hyper-parameter to the step-size algorithm.

Algorithm 4.3.1: damped Adaptive Stochastic Gradient Descent Biased

Input: $x_0 \in \mathbb{R}^d, \lambda_0 > 0, \theta_0 = +\infty, \xi^0, \alpha > 0, 1 > \beta > 0$

Routine:

$k := 0$

$x^1 = x_0 - \lambda_0 \nabla f(x_0)$

while *stopping criterion is not satisfied* **do**

  Sample ξ^k

  $L_k := \frac{\|\nabla f_{\xi^k}(x^k) - \nabla f_{\xi^k}(x^{k-1})\|}{\|x^k - x^{k-1}\|}$

  $\lambda_k := \min\left\{\sqrt{1+\beta\theta_{k-1}}\lambda_{k-1}, \frac{\alpha}{L_k}\right\}$

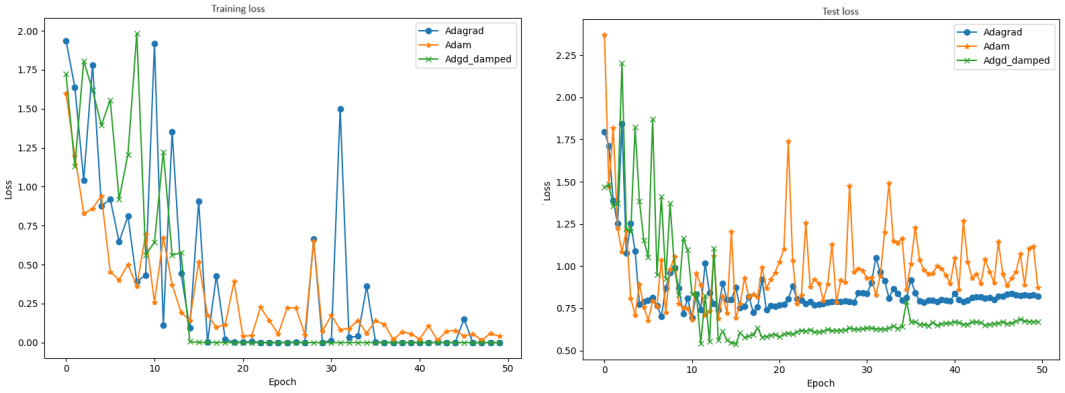
  $x^{k+1} := x^k - \lambda_k \nabla f_{\xi^k}(x^k)$

  $\theta_k := \frac{\lambda_k}{\lambda_{k-1}}$

  $k = k + 1$

This idea was already used in the original paper [9] and a value for $\beta = 0.02$ was used. Applying this new version of the algorithm to our problem, we get the following results. We see that this helped a lot and we were able to outperform Adam and Adagrad.

Figure 4.6: Loss for training and test data after damping



As we can see in the Figure 4.7, the new parameter helped keeping the learning rate in bounds. Looking at the plot for the losses again in a logarithmic scale we see that the algorithm reached a lower loss and therefore yielded a better loss on the test set which improved accuracy by almost 5% as can be seen in Figure 4.8. It should be noted that decreasing α did not have a similar effect on the performance ,therefore the new hyperparamter was indeed needed.

Figure 4.7: Damped gradient

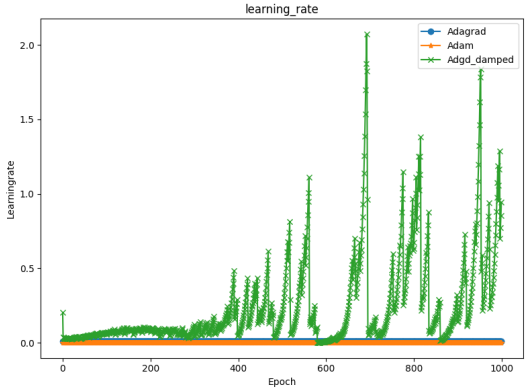
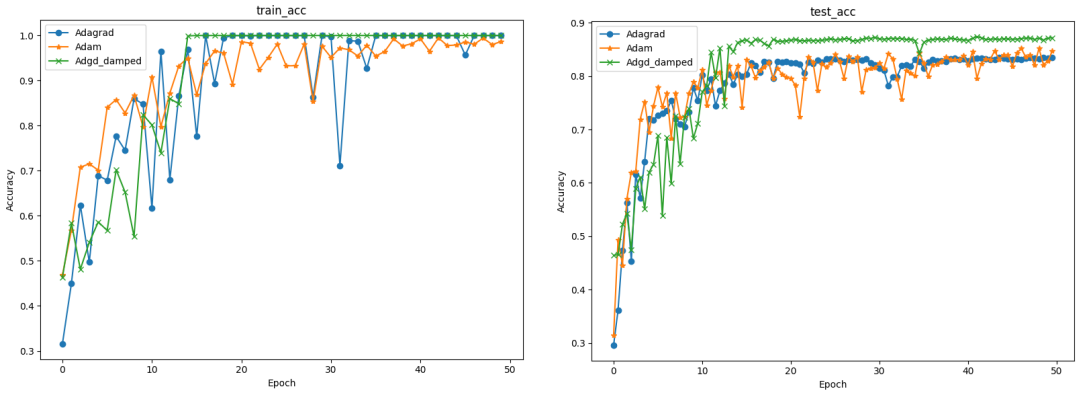


Figure 4.8: Accuracy for training and test data



Bibliography

[1] Andersen Ang. *Nesterov's accelerated gradient descent on L-smooth convex function.* `https://angms.science/doc/CVX/CVX_NAGD.pdf`, 2023.

[2] Larry Armijo. *Minimization of functions having Lipschitz continuous first partial derivatives.* Pacific Journal of Mathematics, Vol. 16(No. 1):1 – 3, 1966.

[3] Michel Bierlaire. *Optimization: Principles and Algorithms.* EPFL Press, Lausanne, 2nd edition, 2018.

[4] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. *Optimization Methods for Large-Scale Machine Learning.* `https://arxiv.org/abs/1606.04838`, 2016.

[5] Dheeru Dua and Casey Graff. *UCI* university of california, irvine, machine learning repository. `https://archive.ics.uci.edu/datasets`, 2017.

[6] Roger Fletcher. *Practical methods of optimization: Unconstrained optimization v. 1.* A wiley-interscience publication, England: John Wiley and Sons., 1980.

[7] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. *CIFAR-10* (canadian institute for advanced research). `http://www.cs.toronto.edu/~kriz/cifar.html`.

[8] Claude Lemarechal. *Optimization and Optimal Control.* Springer, Berlin, Heidelberg, 1981.

[9] Yura Malitsky and Konstantin Mishchenko. *Adaptive Gradient Descent without Descent.* `https://arxiv.org/abs/1910.09529`, 2019.

[10] Maximilian Janisch. *Equivalent definitions of convexity*, Mathematics Stack Exchange. `https://math.stackexchange.com/q/3996184`, 2021.

[11] Yurii Nesterov. *A method of solving a convex programming problem with convergence rate* $O\left(\frac{1}{k^2}\right)$. `https://www.mathnet.ru/eng/dan46009`, 1983.

[12] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Springer, 1st edition, 2004.

[13] Lam M. Nguyen, Phuong Ha Nguyen, Marten van Dijk, Peter Richtárik, Katya Scheinberg, and Martin Takáč. *SGD and Hogwild! Convergence Without the Bounded Gradients Assumption.* `https://arxiv.org/abs/1802.03801`, 2018.

[14] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, Vol. 12:2825–2830, 2011.

[15] Sebastian Ruder. *An overview of gradient descent optimization algorithms.* `https://arxiv.org/abs/1609.04747`, 2017.

[16] Wenyu Sun and Ya-Xiang Yuan. *Optimization theory and methods: Nonlinear Programming.* Springer, 2006.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. *Going Deeper with Convolutions.* `https://arxiv.org/abs/1409.4842`, 2014.

[18] Philip Wolfe. *Convergence Conditions for Ascent Methods.* SIAM Review, Vol. 11(No. 2):226–235, 1969.